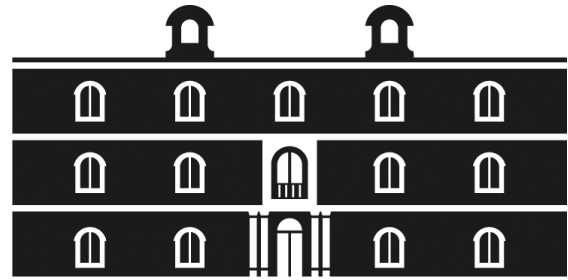




Universidad
Politécnica
de Cartagena



industriales
etsii UPCT

Control de posición y seguimiento de trayectorias para un robot neumático de estructura paralela

Titulación: Ingeniería Industrial
Intensificación: Sistemas Eléctricos
Alumno/a: José Miguel López Sánchez
Director/a/s: Miguel Almonacid Kroeger
José Manuel Cano Izquierdo

Cartagena, 18 de Octubre de 2012

ÍNDICE

CAPÍTULO 1: Introducción.

1.1. Introducción a la historia del control automático y los robots paralelos.	3
1.1.1. Breve historia del control automático.	3
1.1.2. Robots de estructura paralela.	5
1.2. Objetivos del proyecto.	8
1.2.1. Origen.	8
1.2.2. Antecedentes.	8
1.2.3. Objetivos.	10
1.3. Organización de la documentación.	10
1.3.1. Memoria descriptiva.	10
1.3.2. CD-Rom.	11

CAPÍTULO 2: Componentes.

2.1. Introducción.	15
2.2. Tarjeta de adquisición de datos.	16
2.3. Cilindros normalizados de doble efecto.	20
2.4. Convertidores <i>DADE-MVC-010</i> .	21
2.5. Válvulas de control proporcional <i>MPYE-5-1/8-LF-010-B</i> .	25
2.6. Cilindro de doble efecto con vástago pasante.	26
2.7. Transmisor de posición <i>SMAT-8E-S50-IU-M</i> .	27
2.8. Válvulas digitales <i>MOFH-3-1/8</i> .	28
2.9. Válvula reguladora de presión <i>LFR-D-MINI</i> .	29
2.10. Fuente de alimentación.	30
2.11. Circuitos de potencia.	30
2.12. Otros elementos utilizados.	31

CAPÍTULO 3: Software y desarrollo del programa informático.

3.1. Software utilizado.	35
3.1.1. Descripción y motivos de su uso.	35
3.1.2. Herramientas utilizadas.	36
3.2. Desarrollo del programa informático.	37
3.2.1. Programas para la puesta en marcha.	38
3.2.1.1. Recibir datos de sensores.	38
3.2.1.2. Establecimiento del punto de referencia.	38
3.2.2. Control de la posición de los cilindros <i>DNCL</i> .	39
3.2.2.1. Función <i>PID</i> .	39
3.2.2.2. Función <i>CALLBACK_PID</i> .	40
3.2.2.3. Función <i>FINALIZA</i> .	44
3.2.3. Cinemática del robot.	45
3.2.3.1. Cinemática directa, <i>CDIR</i> .	46
3.2.3.2. Cinemática inversa, <i>CINV</i> .	47

CAPÍTULO 4: Fundamentos de control y puesta en marcha del sistema.

4.1. Introducción.	51
4.2. Puesta en marcha.	51
4.2.1. Referencia y calibración de los cilindros.	51
4.2.2. Problemas tras el encendido.	54
4.3. Elección de la frecuencia de muestreo.	55
4.4. Elección de la presión de suministro de aire comprimido.	58
4.5. Introducción al control mediante PID.	60
4.5.1. Acciones básicas del control PID.	60
4.5.2. Sintonía de controladores PID.	61

CAPÍTULO 5: Análisis del movimiento desacoplado de los cilindros.

5.1. Introducción.	65
5.2. Primeras pruebas de desplazamiento.	65
5.2.1. Comparación cilindro 1 y cilindro 2.	68
5.2.2. Comparación a distintas referencias y posiciones iniciales.	69
5.3. Pruebas para obtener la función de transferencia del sistema.	73
5.3.1. Método en lazo abierto.	73
5.3.2. Método en lazo cerrado.	77
5.3.2.1. Método gráfico.	77
5.3.2.2. Mediante entrada vector de pulsos PRBS.	80
5.4. Simulaciones gráficas de las funciones de transferencia obtenidas.	82
5.4.1 Comparaciones entre métodos.	82
5.4.2. Comparaciones entre cilindros.	83
5.5. Sintonización de los controladores PID.	84
5.5.1. Sintonización experimental de controladores.	85

CAPÍTULO 6: Control simultáneo de los cilindros.

6.1. Introducción.	93
6.2. Perturbaciones asociadas al desplazamiento de los cilindros.	93
6.3. Sintonización de los controladores PID para el movimiento simultáneo.	99
6.4. Sincronización y saturación del caudal de aire en las válvulas.	102

CAPÍTULO 7: Seguimiento de trayectorias.

7.1. Introducción.	105
7.2. Tratamiento de coordenadas y espacio de trabajo.	105
7.2.1. Tratamiento de coordenadas.	105
7.2.2. Espacio de trabajo.	106
7.3. Figuras realizadas.	108
7.3.1. Cuadrado por coordenadas manuales.	108
7.3.2. Polígonos regulares.	113.
7.3.2.1. Función <i>polígono.m</i> .	113
7.3.2.2. Pruebas con distinto número de lados.	114
7.3.3.3. Aproximación al círculo.	124
7.3.3. Senoide.	126

CAPÍTULO 8: Conclusiones y desarrollos posteriores.

8.1. Conclusiones.	133
8.2. Desarrollos futuros.	134

BIBLIOGRAFÍA.	137
----------------------	-----

CAPÍTULO 1: **INTRODUCCIÓN.**

1.1. Introducción a la historia del control automático y los robots paralelos.

En el presente proyecto se va a controlar la posición de un robot de estructura paralela con acción neumática. El control es la acción y el efecto de poder decidir sobre el desarrollo de un proceso o sistema para que actúe de la forma deseada. Un sistema automatizado se define como aquél capaz de reaccionar sin intervención de un operario ante cambios que se producen en el mismo, dando lugar a las acciones adecuadas para cumplir la función para la que ha sido diseñado.

1.1.1. Breve historia del control automático.

El control automático ha sufrido un rápido desarrollo en los últimos años desde que, aproximadamente en el año 300 a.C., los griegos empezaran a tener la necesidad de medir empíricamente el tiempo. Esta evolución ha continuado con etapas tan importantes como la Revolución Industrial y las Guerras Mundiales.

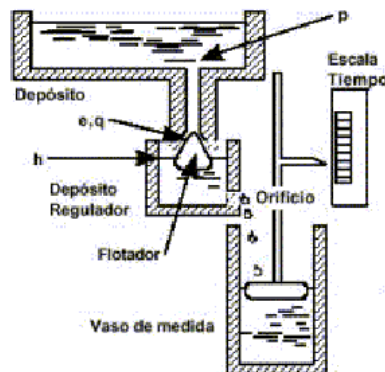


Fig. 1.1. Reloj de agua de Ktesibios (siglo III a.C.)

Cuando la máquina de vapor fue inventada y se inició la Revolución Industrial, paralelamente apareció la necesidad de realizar un sistema de control para manipular los diferentes parámetros de esta máquina. De este modo se desarrollaron dispositivos de control tales como reguladores flotantes, reguladores de temperatura, reguladores de presión y dispositivos de control de velocidad.

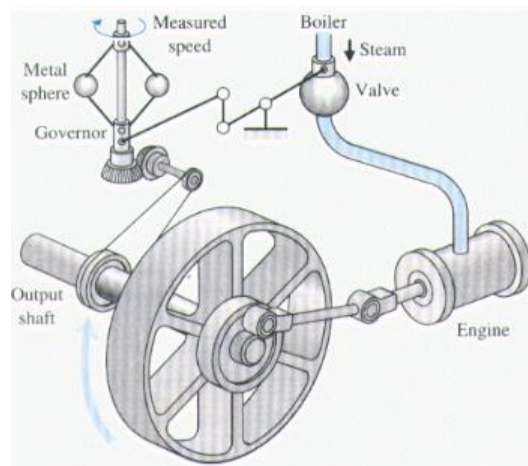


Fig. 1.2. Regulador centrífugo de Watt (1788)

Hasta finales del siglo XIX el Control Automático era prácticamente intuitivo. El deseo de mejorar las respuestas y la exactitud de los sistemas de control obligó a desarrollar la teoría del control. Aparecieron entonces los primeros análisis de Control Automático, basados en el regulador centrífugo. Se desarrollaron los conceptos de estabilidad, modelos matemáticos simples, linealización, la importancia de la acción integral y los criterios de estabilidad.

Tanto en la Primera como en la Segunda Guerra Mundial, la necesidad de implementar diferentes controles para barcos, aviones y demás, motivó y, a su vez, provocó que un gran número de personas se involucraran en el desarrollo de los mismos, dando lugar a un gran avance del Control Automático en estas etapas. Los métodos de respuesta en frecuencia y el lugar de las raíces de Evans (1950) dieron paso al análisis de sistemas complejos en el dominio del tiempo.



Fig. 1.3. Los laboratorios Bell fueron uno de los organismos más influyentes en el desarrollo de los sistemas de control

Desde el año 1957 en que comienza la era espacial y de la informática, y hasta nuestros días, el desarrollo tecnológico ha permitido que el Control Automático sea algo cotidiano que busca facilitar la vida de los seres humanos. Actualmente la tendencia de los sistemas de control es hacia la optimización y digitalización total de los controladores.

La automatización de los procesos productivos sigue siendo posible gracias al desarrollo y abaratamiento de la tecnología. Algunas de las ventajas del Control Automático son las siguientes:

- Aumento de la productividad.
- Mejora de la calidad y homogeneización de los productos.
- Reducción de costes por inversión en mano de obra no especializada.
- Flexibilidad del sistema productivo.

Estos aspectos se han convertido en requisitos indispensables para mantener la competitividad, por lo que el aumento del nivel de automatización de los procesos es una necesidad de la sociedad actual.

1.1.2. Robots de estructura paralela.

Un robot paralelo es aquél cuya estructura mecánica está formada por un mecanismo de cadena cerrada en el que el efector final se une a la base por al menos dos cadenas cinemáticas independientes. Además, cada cadena dispone, en general, de un único actuador, reduciendo así su complejidad y permitiendo canalizar mejor la energía de los accionadores hacia mejorar las prestaciones del robot, bien en cuanto a velocidad de movimiento o a capacidad de carga de su efector.

Los mecanismos paralelos que han logrado un mayor reconocimiento general y han contribuido a la aparición de un mayor número de publicaciones sobre la robótica paralela son, por estricto orden temporal, los correspondientes al Dr. Gough (1947), Mr. Stewart (1965), Mr. Cappel (1967) y Hunt (1978).

El Dr. Gough diseñó un octaedro hexápodo con lados de longitud variable (Figura 1.4) como plataforma para la comprobación del comportamiento de los neumáticos de la casa Dunlop bajo cargas aplicadas en diferentes ejes. De esta forma, intentaba simular el proceso de aterrizaje de un avión. En la actualidad existen multitud de plataformas basadas en este diseño, conocidas bajo el nombre de MAST (Multi-Axis Simulation Table).

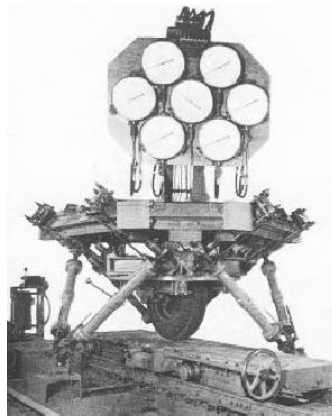


Fig. 1.4. Primera plataforma de Gough

En 1965, Mr. Stewart presentó un artículo en el que describía una plataforma de movimiento de 6 grados de libertad destinada a trabajar como simulador de vuelo (Figura 1.5). Este artículo tuvo y tiene una gran influencia en el mundo académico y se considera uno de los primeros trabajos de análisis de plataformas paralelas.

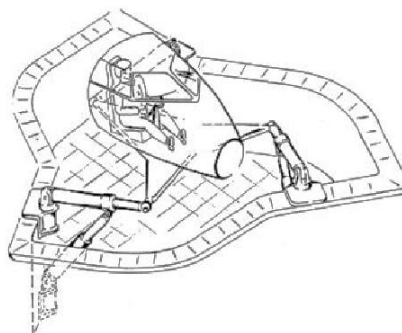


Fig. 1.5. Plataforma de Stewart

Paralelamente, el ingeniero Klaus Cappel realizaba en el Franklin Institute Research Laboratory numerosas investigaciones con plataformas paralelas de 6 grados de libertad, y en 1967 patentaba un simulador de movimiento basado en un hexápodo (Figura 1.6).

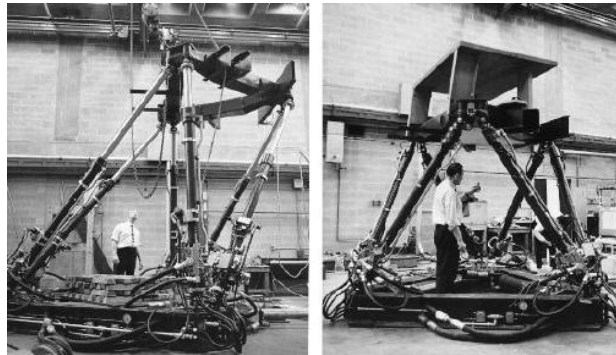


Fig. 1.6. Simulador de movimiento de Klaus Cappel

En 1978, Hunt sugirió que se usaran los mecanismos de los simuladores de vuelo como robots manipuladores y destacó que los manipuladores paralelos requerían de un estudio más detallado en el contexto de las aplicaciones robóticas a la vista de las ventajas en cuanto a rigidez y precisión respecto a los robots serie convencionales.

La robótica paralela está sufriendo actualmente un avance más rápido que en épocas anteriores, fundamentalmente gracias al aumento de la capacidad de cómputo de los nuevos procesadores para resolver procesos numéricos iterativos, y a un mejor conocimiento del funcionamiento de estas estructuras. Este tipo de mecanismos son utilizados especialmente en determinadas aplicaciones en las que sus características ofrecen ventajas para resolver ciertos problemas para los que los robots serie tienen limitaciones.

En general, las plataformas paralelas presentan las siguientes ventajas:

- Los accionamientos de potencia conectan directamente la base del robot al efector final. Así, éstos sirven de elementos estructurales y actúan de manera simultánea, lo que les da la capacidad de manipular cargas muy superiores a su propio peso. Por tanto, el elevado ratio carga/potencia de estos mecanismos proporciona una alta eficiencia energética.
- Las estructuras paralelas son mecanismos que ofrecen una alta rigidez y muy bajo peso. Esto hace que presenten unas características en cuanto a precisión claramente superiores a las de los robots serie.
- Presentan elevadas velocidades de operación, en comparación con cualquier otro tipo de estructura robótica.

Sin embargo, también cabe destacar otras características que, según la aplicación, pueden considerarse desventajas:

- La cinemática de los mecanismos paralelos es más complicada, lo que en ocasiones obliga a recurrir a sensores redundantes para poder establecer un lazo de control.

- El espacio de trabajo suele ser pequeño comparativamente hablando, y su cálculo no es sencillo debido a que la posición y orientación están fuertemente acopladas.
- El problema de las configuraciones singulares es más complejo, y debe resolverse específicamente para cada tipología.
- No existe, como ocurre en los robots serie, un modelo dinámico general para los mismos. Esto dificulta el desarrollo de algoritmos de control de carácter general y hace que los robots existentes en la actualidad se controlen de forma desacoplada.

Según lo expuesto en este último punto, el tema del control en este tipo de robots es un campo abierto y los trabajos aportados hasta la fecha no son muy rigurosos. Los sistemas de control utilizados en los robots actualmente son de tipo desacoplado formados por un modelo cinemático que genera referencias para las articulaciones y los servos de éstas. En este esquema, la solución de la dinámica inversa en robots paralelos simplifica, respecto a los robots serie, la implantación eficiente de los sistemas de control.

Uno de los trabajos realizados en este campo es el de Liu y otros (*Liu et al., 1992*) que discutieron varios aspectos prácticos en relación con el control en el espacio de la tarea del robot o en el espacio articular. Sin embargo, la aplicación de su esquema de control en tiempo real está en duda debido a las intensas tareas computacionales implicadas en la solución de la cinemática directa. Una alternativa preferible es planificar la trayectoria en el espacio cartesiano, que es más natural en los robots paralelos y que sólo requiere la cinemática inversa. Esta alternativa será la desarrollada en el presente proyecto para controlar la posición del efector final.

La gran variedad de topologías existentes provoca que el campo de aplicación de estos robots sea muy amplio, y abarca desde microbots posicionadores hasta grandes plataformas de gran capacidad de carga, desde aplicaciones médicas hasta simuladores de vuelo, desde dispositivos hápticos a robots manipuladores, desde robots experimentales a máquinas herramientas de control numérico.

Los requerimientos en cuanto a versatilidad en el posicionamiento de las herramientas en los centros de mecanizado hacen de gran interés el uso de mecanismos paralelos con este fin (Figura 1.7).



Fig. 1.7. Centro de mecanizado Tornado de la compañía Hexel

1.2. Objetivos del proyecto.

1.2.1. Origen.

El nacimiento de este proyecto es fruto de la iniciativa del Departamento de Ingeniería de Sistemas y Automática. La idea inicial era construir una maqueta de uso didáctico para controlar la posición de una plataforma mediante dos cilindros neumáticos de doble efecto accionados mediante electroválvulas. Con ella se pretendían realizar distintas prácticas, según el área de conocimiento que se requiera.

Por ejemplo, en el área de control, permitiría las siguientes aplicaciones:

- Modelado y simulación de sistemas electroneumáticos.
- Control de posición de un cilindro neumático (SISO).
- Control de posición de la plataforma (MIMO).
- Aplicación de distintas técnicas de control (fuzzy, adaptativo, predictivo).
- Control de sistemas no lineales.

En el campo de la automatización, se podría estudiar:

- Sensores y actuadores electroneumáticos.
- Control con autómatas de un sistema con entradas y salidas analógicas.

Por último, en el área de la robótica, se pretendía realizar:

- Modelado y simulación de un robot paralelo neumático de dos grados de libertad.
- Control acoplado.
- Control de trayectorias.

1.2.2. Antecedentes.

El proyecto *Robowriter* fue concebido con la finalidad de describir trayectorias gráficas sobre un papel. No obstante, con el fin de cubrir diferentes aplicaciones, se construyó un robot de alta flexibilidad, es decir, fácilmente transportable y que puede ser utilizado en posición horizontal o vertical. De este modo, gracias a su versatilidad se pueden realizar otras tareas simplemente cambiando el elemento soporte para la escritura por otro adaptado a una herramienta determinada.

El diseño de *Robowriter* se inició con el proyecto realizado en 2006 por el alumno Miguel Torregrosa Puig, en el que se incluían el modelado, simulación y diseño del robot. Este proyecto cubría los siguientes objetivos:

De la parte de modelado:

- Simulación gráfica del robot.
- Planteamiento y resolución de la cinemática directa.
- Planteamiento y resolución de la cinemática inversa.
- Representación de los resultados obtenidos en el cálculo del espacio de trabajo.

De la parte correspondiente a la simulación:

- Estudio y características de las piezas proporcionadas por el fabricante.
- Representación de las posibles posiciones del robot en *Matlab*.
- Ensamblaje, planteamiento de las restricciones geométricas y físicas entre piezas y animación gráfica con restricciones de movimiento en el programa *Catia*.

De la parte de diseño:

- Diseño de las piezas necesarias para el funcionamiento del robot.
- Diseño de la estructura que incorpora el robot y elementos necesarios.
- Diseño básico de la instalación neumática y de control.

Teniendo como base el proyecto anteriormente mencionado, en el año 2008 el alumno Alejandro Rosillo Meseguer desarrolló el montaje físico del robot y su puesta en marcha. Los objetivos planteados en su momento fueron los siguientes:

- Conexión eléctrico entre la tarjeta de adquisición de datos, los circuitos de adaptación de tensión, las válvulas y los cilindros.
- Realización del montaje físico de todas las partes que compondrán el robot.
- Programación de las tarjetas de adquisición.
- Programación de los algoritmos de control y de un interface mediante un *GUI* realizado en *Matlab*.
- Calibración y puesta en marcha de todo el robot paralelo.
- Averiguar el grado de efectividad que pueda llegar a tener el control todo-nada en procesos que varíen rápidamente y requieran cierta precisión.
- Control de posición de un cilindro neumático (SISO: Single Input Single Output) y del robot (MIMO: Multiple Input Multiple Output).
- Aplicación de distintas técnicas experimentales de sintonía de controladores PID.

Finalmente, en el año 2009 el alumno Ginés Moreno Martínez continuó desarrollando el control del robot paralelo, consiguiendo los siguientes objetivos:

- Puesta a punto de sensores y actuadores, y revisión del conexionado eléctrico y neumático.
- Realización de unos nuevos circuitos adaptadores de tensión para las válvulas *MOFH*.
- Prueba de unos PID iniciales a varias presiones de funcionamiento.
- Obtención de un modelo matemático de los cilindros por métodos experimentales.
- Sintonización de unos controladores PID de cada cilindro individual (SISO) y sintonización de controladores para una respuesta conjunta del sistema (MIMO).
- Introducción de la cinemática directa e inversa en los programas de control del robot para poder dibujar figuras geométricas básicas.

1.2.3. Objetivos.

Partiendo del estudio y análisis de los proyectos descritos anteriormente, para el presente proyecto se han fijado los siguientes objetivos:

- Revisión del conexionado eléctrico y neumático. Instalación de los componentes faltantes.
- Elección de la presión de aire de suministro y de la frecuencia de muestreo para la puesta en marcha del sistema.
- Implementar controladores PID para el movimiento desacoplado del robot (SISO).
- Extracción de funciones de transferencia para obtener un modelo matemático de cada cilindro.
- Análisis del movimiento acoplado del robot (MIMO). Corrección de perturbaciones.
- Implementar uno o varios controladores PID que permitan al sistema describir trayectorias gráficas sobre un papel.
- Programación de figuras patrón para el control de posición del robot.
- Programación del seguimiento de trayectorias de la plataforma del robot.

1.3. Organización de la documentación.

La documentación del proyecto se presenta en dos formatos, la memoria descriptiva en formato papel y un CD-Rom.

1.3.1. Memoria descriptiva.

La presente memoria resume tanto los planteamientos teóricos como los resultados prácticos obtenidos durante el desarrollo del proyecto. El capítulo 1 da a conocer el proyecto a realizar y un breve recorrido histórico por los procesos de control automático y los robots de estructura paralela. En el segundo capítulo se describen todos y cada uno de los componentes eléctricos y neumáticos que conforman el robot, así como las conexiones entre ellos. En el capítulo 3 se desarrolla el software utilizado y los programas informáticos que permiten operar con el robot. En el capítulo cuarto se muestran las primeras pruebas realizadas para establecer los parámetros previos a la puesta en marcha del sistema y los fundamentos de control que se aplicarán en los siguientes capítulos. El quinto capítulo desarrolla la sintonización experimental de los controladores PID mediante el análisis desacoplado del sistema. Del mismo modo, en el sexto capítulo se exponen los procedimientos destinados a la obtención de unos controladores PID para el movimiento conjunto del sistema y se explica el método empleado para reducir las perturbaciones que se producen entre los cilindros. En el capítulo 7 se presentan los resultados obtenidos de realizar el seguimiento de la trayectoria correspondiente al contorno de distintas figuras geométricas. Por último, en el capítulo 8 se exponen las conclusiones del proyecto y posibles vías futuras para una mejora del comportamiento del robot.

La memoria descriptiva incluye además un índice, la bibliografía empleada y los anexos con los planos eléctricos y neumáticos del sistema, los problemas surgidos y su resolución y los *scripts* de *Matlab* empleados.

1.3.2. CD-Rom.

En este soporte informático se incluye:

- La memoria descriptiva y los anexos en formato electrónico.
- Los *scripts* y funciones de *Matlab* que conforman el código necesario para la puesta en marcha y el desplazamiento del robot.
- Las diapositivas para la presentación del presente proyecto.
- Los datos y gráficas de *Matlab* obtenidas en pruebas de control en el Laboratorio.
- Manuales de fabricante de cada una de los elementos del robot.
- Fotografías de las partes del robot.
- Videos del funcionamiento del robot.

CAPÍTULO 2:

COMPONENTES.

2.1. Introducción.

En este capítulo se describen los elementos de los que consta el robot de estructura paralela (figura 2.1), así como las conexiones requeridas para el correcto funcionamiento del sistema y otros detalles de interés en el uso de los componentes.

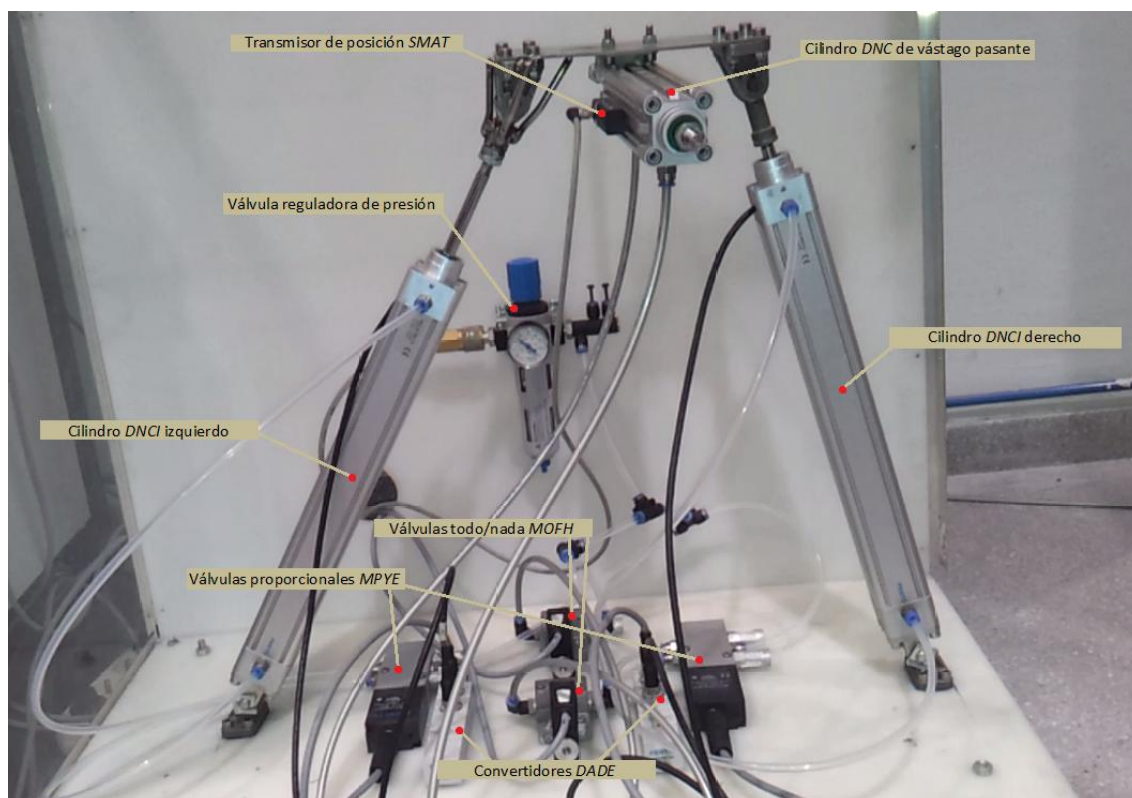


Fig. 2.1: Robowriter y sus componentes

2.2. Tarjeta de adquisición de datos.

La tarjeta empleada en el presente proyecto es la *PCI-MIO-16E-1* de *National Instrument* (figura 2.2). Se conecta al PC mediante un puerto PCI que además proporciona la tensión necesaria para la alimentación de la propia tarjeta. Los controladores necesarios para su correcto funcionamiento (*ni-daq 7.3*) vienen incluidos al adquirir el producto y se encuentran instalados en el ordenador que gobierna el sistema.

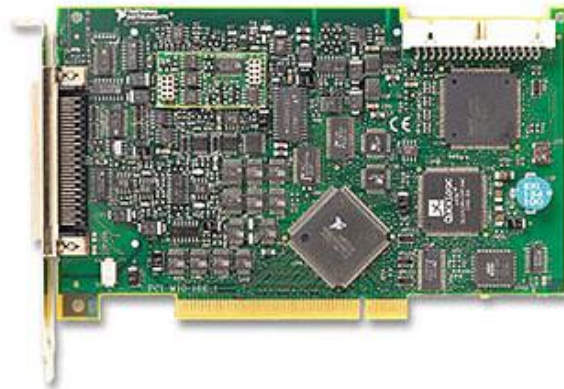


Fig. 2.2: PCI-MIO-16-E-1 National Instrument

Para la transmisión de datos entre la tarjeta y los elementos del robot se emplea la bornera *CB68LP* (figura 2.3), conectada al PC mediante el cable de 68 pines *R6850 08270201 L* (figura 2.4).

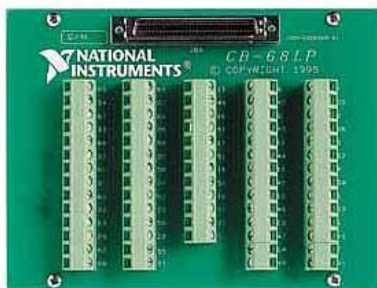


Fig. 2.3: Bornera CB68LP



Fig. 2.4: Cable 68 pines R6850 08270201 L

La conexión de las señales analógicas puede variar según la naturaleza de la fuente de alimentación y el tipo de tarjeta de adquisición de datos.

Las entradas analógicas pueden conectarse de tres formas distintas:

- Diferencial.
- RSE (Referenced single ended).
- NRSE (Non-Referenced single ended).

La fuente de alimentación usada en este proyecto es referenciada, es decir, está conectada a la red eléctrica y tienen masa propia. En este caso, el orden de preferencia para usar un tipo de entrada u otra es el siguiente:

1. Diferencial: la ventaja que presenta este modo es que aísla el ruido en modo común, por lo que es el método más seguro y que mejor resultados ofrece. El inconveniente es que requiere dos entradas para cada medida, lo cual no supone un problema real en este caso puesto que la bornera utilizada dispone de canales de entrada suficientes para hacer frente a todos los componentes del sistema.
2. NRSE: como el equipo a medir tiene una referencia propia no se debe conectar ésta al GND de la tarjeta, ya que son referencias distintas y existe un potencial entre ellas. Se debe conectar el positivo de la señal al canal correspondiente y el negativo a AI SENSE, que es un punto común de la tarjeta pero no es masa.
3. RSE: no se debe usar en ningún caso, ya que se interconectarían entre sí masas distintas.

Según lo expuesto anteriormente, para realizar las conexiones de los distintos componentes se seguirán las siguientes pautas:

- Se emplearán 6 entradas (AI) y 2 salidas analógicas (AO), y 3 terminales de entrada/salida digital (DIO).
- Las señales analógicas se conectarán en modo diferencial para una mayor supresión del ruido, precisando 2 cables para cada medida.
- Se conectará el terminal positivo al pin analógico *AI n* y el negativo, al *AI n+8*. Asimismo, éste se unirá con los terminales de referencia o masa (*AI GND*) para que sean comunes a todo el sistema.
- Se utilizarán dos entradas analógicas para el convertidor de cada cilindro *DNCL* y para el transmisor de posición del cilindro con vástago pasante.
- Las dos salidas analógicas corresponden a las dos válvulas de control proporcional, de forma que el terminal positivo irá al terminal *AO n* y el negativo a *AO GND*.
- Las señales digitales corresponden a las dos válvulas de control todo/nada y al terminal de referencia y calibración de los convertidores *DADE*.

Finalmente, la bornera debe quedar tal y como se observa en la figura 2.5, empleando los terminales marcados en la figura 2.6, con los cuáles se ha programado el control del robot. Es importante destacar que si se modifica el pin para alguna conexión es necesario realizar la corrección correspondiente en el código de programación.

Nótese, en la figura 2.6, que el pin correspondiente a la entrada/salida digital DIO2 no ha sido utilizado debido a que no proporcionaba un valor de tensión suficiente para activar las entradas de referencia y calibración de los convertidores. En su lugar se ha usado el terminal DIO3.

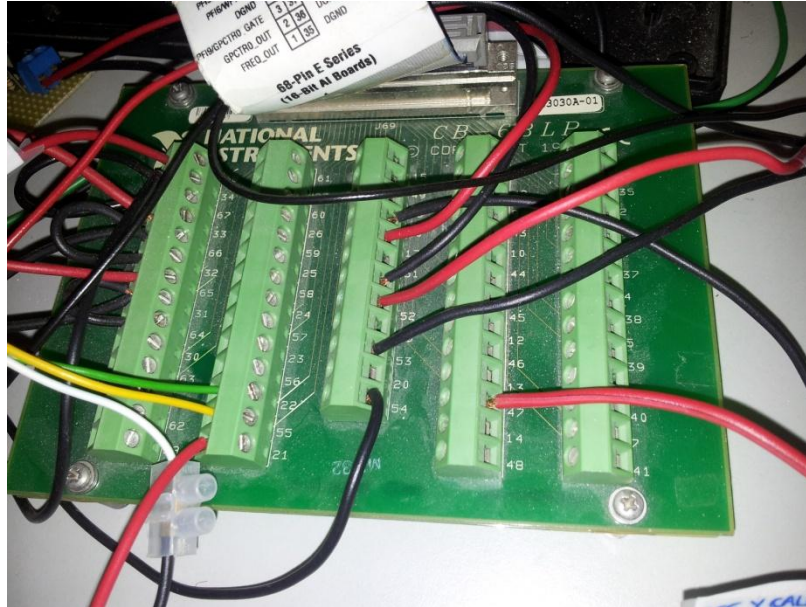


Fig. 2.5: Conexiones en la bornera

AI 8 ●	34	68	AI 0 ●
AI 1 ●	33	67	AI GND ●
AI GND ●	32	66	AI 9 ●
AI 10 ●	31	65	AI 2 ●
AI 3	30	64	AI GND ●
AI GND	29	63	AI 11
AI 4	28	62	AI SENSE
AI GND	27	61	AI 12
AI 13	26	60	AI 5
AI 6	25	59	AI GND
AI GND	24	58	AI 14
AI 15	23	57	AI 7
AO 0 ●	22	56	AI GND
AO 1 ●	21	55	AO GND ●
AO EXT REF	20	54	AO GND ●
P0.4	19	53	D GND ●
D GND ●	18	52	P0.0 ●
P0.1 ●	17	51	P0.5
P0.6	16	50	D GND ●
D GND	15	49	P0.2
+5 V	14	48	P0.7
D GND	13	47	P0.3 ●
D GND	12	46	AI HOLD COMP
PFI 0/AI START TRIG	11	45	EXT STROBE
PFI 1/AI REF TRIG	10	44	D GND
D GND	9	43	PFI 2/AI CONV CLK
+5 V	8	42	PFI 3/CTR 1 SRC
D GND	7	41	PFI 4/CTR 1 GATE
PFI 5/AO SAMP CLK	6	40	CTR 1 OUT
PFI 6/AO START TRIG	5	39	D GND
D GND	4	38	PFI 7/AI SAMP CLK
PFI 9/CTR 0 GATE	3	37	PFI 8/CTR 0 SRC
CTR 0 OUT	2	36	D GND
FREQ OUT	1	35	D GND

Fig. 2.6: Terminales utilizados de la bornera

2.3. Cilindros normalizados de doble efecto.

Se emplearán dos cilindros del modelo *DNCI-32-300-P-A* (figura 2.7), cuya función será desplazar la estructura sobre el plano vertical de trabajo.

Cada cilindro se fija a la base de la estructura mediante cuatro tornillos en los agujeros roscados de la culata y un pasador que permita el giro. Ambos se unen por la parte superior mediante una pieza que soporta un tercer cilindro en posición horizontal. El cilindro izquierdo va roscado a dicha pieza para mantener un ángulo fijo de 120° mientras que el derecho se sujeta por medio de una rótula que permite un giro relativo entre el cilindro y la pieza de unión.

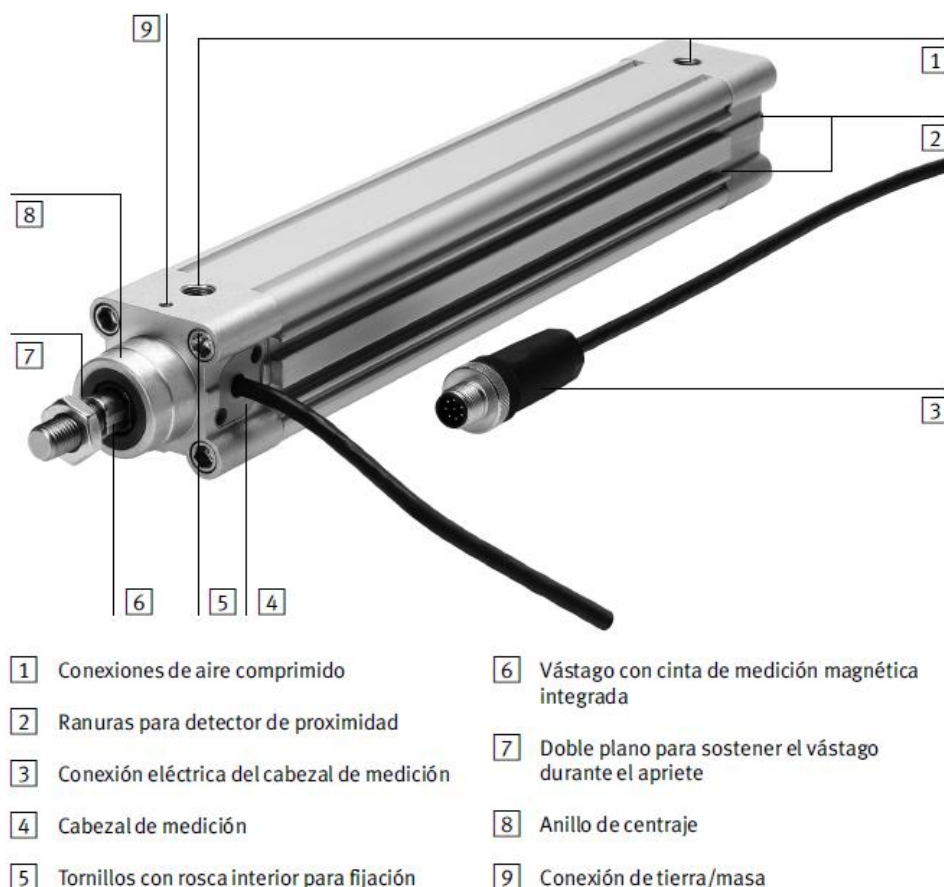


Fig. 2.7: Cilindro DNCI-32-300-P-A

Estos cilindros incorporan un transductor de posición que junto a las válvulas proporcionales y los convertidores permitirá controlar el desplazamiento de los vástagos. Los sensores generan una onda senoidal y otra cosenoidal con distinta fase. Esta diferencia de fase será analizada por los convertidores *DADE-MVC-010* para obtener una salida lineal de 0 a 10 V. De este modo se puede usar la señal de tensión directamente, prescindiendo de una conversión por software que ralentizaría el proceso y obligaría a aumentar el número de entradas analógicas a la tarjeta de adquisición de datos.

2.4. Convertidores *DADE-MVC-010*.

Se trata de un dispositivo que transforma la señal del sensor del cilindro en función del desfase entre ángulos a una señal analógica en intensidad o tensión (0 - 10 V) en este caso. Su esquema de funcionamiento se observa en la figura 2.8. Para el proyecto que atañe esta memoria no se usará un PLC, sino una tarjeta de adquisición da datos.

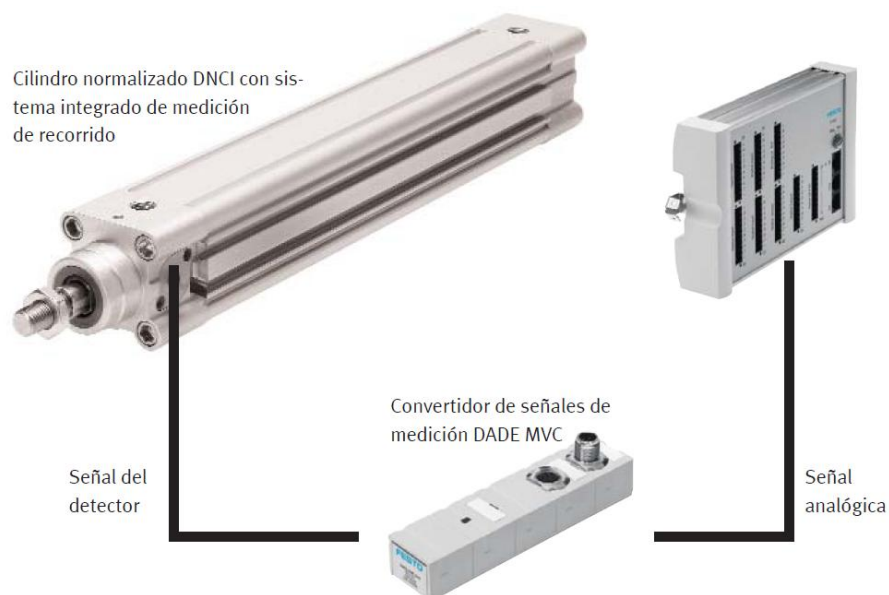


Fig. 2.8: Esquema funcionamiento convertidor *DADE-MVC*

Se emplearán dos convertidores, uno para cada cilindro *DNCI*. La conexión entre el sensor y el convertidor se efectuará directamente mediante el terminal indicado como S2, y mediante el terminal S1 se conectará el convertidor a la bornera (ver figura 2.9).

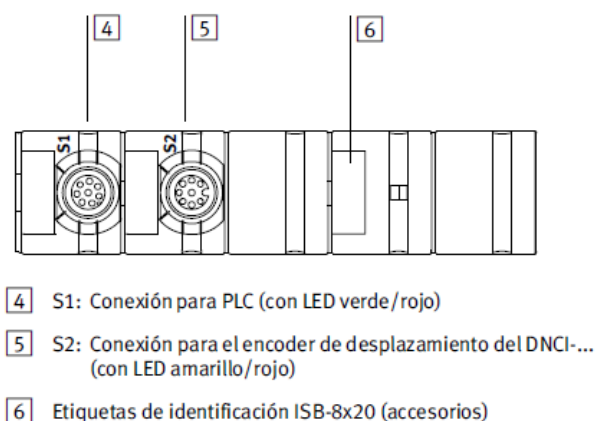


Fig. 2.9: Conexiones del convertidor *DADE-MVC-010*

El cable *S1* está compuesto internamente por 8 cables (figura 2.10), cada uno con una función determinada (figura 2.11).



Fig. 2.10: Cable *S1* de conexión entre el convertidor y la bornera

Pin	Color	Asignación	Conexión S1
1	blanco	+24 V alimentación	
2	marrón	Señal medida (analógica)	
3	verde	Salida de referencia	
4	amarillo	0 V de la señal medida	
5	gris	Entrada de referencia	
6	rosa	Entrada de calibración	
7	azul	Salida Ready (preparado)	
8	rojo	0 V alimentación y entradas/salidas	

Fig. 2.11: Asignaciones de los cables del terminal *S1*

El cable de color blanco irá conectado al terminal positivo de la fuente, a 24 V.

El cable marrón es el valor medido, que varía entre 0 y 10 V. Se conectarán a los terminales AI0 y AI1 de la bornera, uno para cada convertidor.

El cable amarillo corresponde a los 0 V de la señal medida, por lo que irá unido a la masa de la fuente de alimentación y a los terminales AI8 y AI9 (uno para cada convertidor),

El cable gris sirve para establecer el punto de referencia, esto es, el punto del recorrido del cilindro para el que el convertidor dará una salida de 0 V. Esta señal se pierde tras una desconexión eléctrica por lo que habrá que referenciar ambos cilindros cada vez que se conecte la fuente de alimentación.

El cable rosa es análogo al gris. Sirve para establecer el punto de calibración, esto es, el punto del recorrido del cilindro para el que el convertidor proporciona su valor máximo, 10 V. Éste, a diferencia del cable gris, no se pierde tras una desconexión eléctrica, por lo que se dejará desconectado una vez completada la calibración de los cilindros.

El cable rojo va unido al amarillo y a 0 V, es decir, al terminal negativo de la fuente de alimentación.

Los cables verde y azul no se conectarán, ya que sirven únicamente para comprobar que el cilindro se encuentra referenciado y calibrado, respectivamente.

Para realizar dicha referencia y calibración será necesario un dispositivo que aumente a 24 V la tensión de salida de la tarjeta de adquisición de datos, que es de 10 V. La primera puesta a punto de los convertidores y cilindros hay que llevarla a cabo al instalar los cilindros o si la memoria ha sido restablecida, ya sea por un fallo en el funcionamiento de los sensores o porque se requiera un reajuste de los valores de referencia y calibración. Para ello se deben seguir los siguientes pasos:

- 1) Conectar la tensión de alimentación.
- 2) Mover el *DNCI* a su posición de referencia (punto cero de la carrera de trabajo).
- 3) Activar la señal de entrada de referencia (24 V en el cable gris) durante al menos 0.5 segundos.
- 4) Desactivar la entrada de referencia. Aparecerá una señal de 24 V en el cable verde, y el punto de referencia se guardará.
- 5) Mover el cilindro a la posición final de la carrera de trabajo y detenerlo.
- 6) Activar la señal de entrada de calibración (24 V en el cable rosa) durante al menos 0.5 segundos.
- 7) Desactivar la entrada de calibración. Aparecerá una señal de 24 V en el cable azul, y el punto de calibración se guardará permanentemente.

Al desactivar la tensión de alimentación el punto de calibración se mantiene en memoria, pero el de referencia se pierde, por lo que se necesario repetir el proceso (pasos del 1 al 4) cada vez que se enciende el sistema. Para ello se encuentra implementado un archivo denominado *referencia.m* que establece automáticamente el punto de referencia en ambos convertidores.

En caso de fallo los convertidores llevan un sistema de LEDs que se iluminarán de una u otra forma para ayudar a diagnosticar el problema, según la tabla de la figura 2.12.

LED S1	LED S2	Fallo / Solución	Reposición
Verde	rojo par-padea 1x	Fallo del sensor: Verifique el sensor y el cable	– Desconectar, conectar
Verde	rojo par-padea 2x	Fallo de referencia: El DNCI se ha movido durante Ref = 1. El DNCI debe estar parado. Realice de nuevo la referencia.	– Desconectar, conectar – o entrada de referencia = 1
Verde	rojo par-padea x3	Fallo de calibración: El DNCI se ha movido durante Cal = 1. El DNCI debe estar parado. Realice de nuevo la calibración	– Desconectar, conectar – o entrada de calibración = 1
Verde	rojo par-padea x4	Fallo de incremento: El número de incrementos entre el punto de referencia y el final de la carrera demasiado pequeños (el DNCI no se ha movido). Realice de nuevo la calibración.	– Desconectar, conectar – o entrada de calibración = 1
Rojo	apagado	Subtensión: Verifique la tensión.	– Desconectar, conectar
Verde	rojo	Fallo carrera: Valor de la señal fuera del margen previsto. Verifique los topes o restablezca.	– Desconectar, conectar – o llevar el DNCI al margen permitido – o restablecer (ver cap. 6.1).
Rojo	rojo	Cortocircuito en la salida: eliminar.	– Desconectar, conectar

Fig. 2.12: Diagnóstico de fallos según la iluminación de los LEDs de los convertidores

2.5. Válvulas de control proporcional *MPYE-5-1/8-LF-010-B*.

Se requieren dos válvulas *MPYE-5-1/8-LF-010-B* (figura 2.13) para controlar y desplazar cada uno de los cilindros normalizados de doble efecto. Estas válvulas de 5/3 vías permiten una regulación progresiva del caudal de aire, de forma que sirven tanto para regular la velocidad como para posicionar los cilindros.



Fig. 2.13: Válvula MPYE-5-1/8-LF-010-B

Las válvulas se conectarán a la bornera mediante sendos cables *KMPYE*, compuestos internamente por cuatro cables de colores con una función asignada.

COLOR	ASIGNACIÓN
MARRÓN	24 V alimentación
BLANCO	0 V alimentación
VERDE	Salida analógica AO (0...+ 10 V)
AMARILLO	Masa salida analógica AOGND (0 V)

El cable marrón irá unido al terminal positivo de la fuente de alimentación, y el blanco al terminal negativo.

El cable verde se conectará a la salida analógica AO0 o AO1, uno para cada válvula.

El cable amarillo se conectará a las masas de las salidas analógicas AOGND.

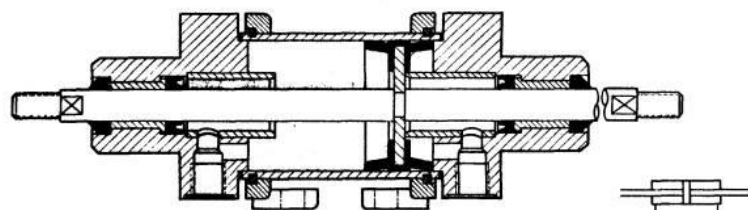
De la hoja de características del dispositivo cabe destacar la tabla de eliminación de fallos (figura 2.14), muy útil a la hora de advertir errores relacionados con la válvula.

Problema	Posible causa	Solución
La corredera no se mueve	Falta tensión de alimentación	Controlar la conexión de 24 V de tensión de alimentación
	Falta valor nominal de referencia	Controlar la unidad de mando, controlar la conexión
	Corredera atascada	Enviar la unidad MPYE a Festo
La corredera vibra	Tensión de alimentación demasiado baja	Procurar la alimentación de 17...30 V c.c.
	Falta apantallamiento del cable o conexión deficiente	Colocar apantallamiento en el extremo del cable alejado de la válvula
	Bucle de conexión a tierra	Aislar la unidad MPYE
Cilindro demasiado lento	Estrangulamiento del paso por error de conexionado (racores basculantes)	Recurrir a conexiones de alternativa

Fig. 2.14: Resumen de posibles fallos de las válvulas MPYE

2.6. Cilindro de doble efecto con vástago pasante.

Se empleará un cilindro *DNC-32-100-PPV-A-S2* (figura 2.15), de vástago pasante. Esto supone que la superficie del émbolo es la misma a ambos lados, por lo que, a igualdad de presión, la fuerza que se ejerce sobre cada lado del émbolo es idéntica, manteniéndose así en reposo. Por este motivo se puede controlar fácilmente con válvulas digitales que introduzcan a cada lado del vástago una presión determinada.

Fig. 2.15: Cilindro *DNC-32-100-PPV-A-S2*

2.7. Transmisor de posición *SMAT-8E-S50-IU-M*.

Para detectar la posición del cilindro de doble efecto se empleará este sensor magnético de posición (figura 2.16).



Fig. 2.16: *SMAT-8E-S50-IU-M8*

La señal de salida puede venir dada en forma de intensidad (4 a 20 mA) o de tensión (0 a 10 V), que será la utilizada en este caso, ya que puede ser leída directamente por la tarjeta de adquisición de datos. Esta señal es siempre proporcional a la posición del vástago (figura 2.17). Si éste se saliera del campo de medición del sensor, se encenderá un led de color rojo.

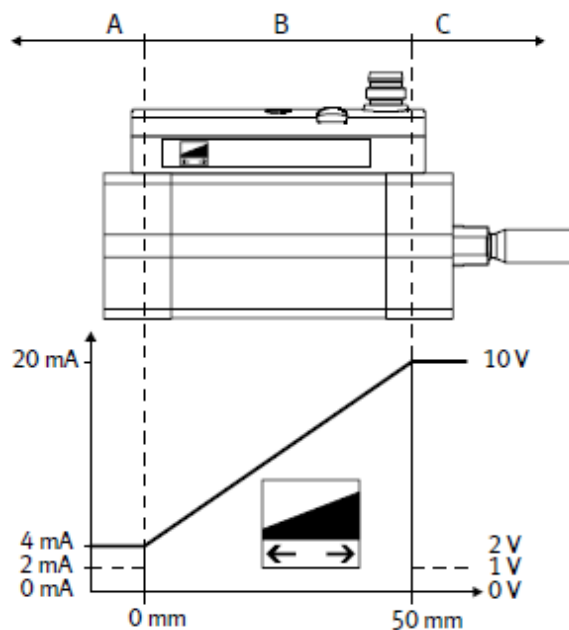
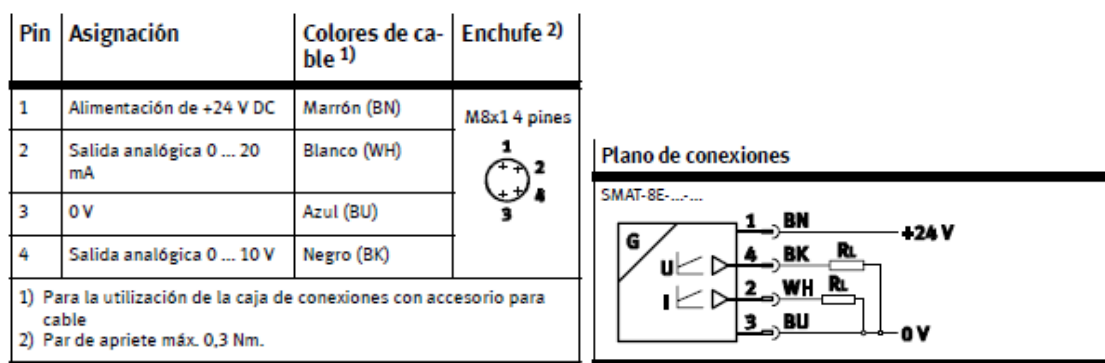


Fig. 2.17: Señal de las salidas analógicas del sensor

La conexión del sensor a la bornera de la tarjeta de adquisición de datos se realizará mediante un cable formado a su vez por cuatro cables internos, con unas funciones definidas (figura 2.18). Tal y como se observa en esta figura, el cable blanco se corresponde con la salida analógica de intensidad, por lo que no será necesario conectarlo.

Fig. 2.18: Conexiones del sensor *SMAT*

El cable marrón va unido al terminal positivo de la fuente de alimentación, y el azul al negativo.

El cable negro proporciona una señal en forma de tensión e irá conectado al terminal AI2 de la bornera.

El cable blanco, que proporciona una señal en forma de intensidad, no será utilizado en este proyecto.

2.8. Válvulas digitales *MOFH-3-1/8*.

Estas válvulas se emplearán para el control todo o nada del cilindro horizontal *DNC*. Se encuentran instaladas dos válvulas de este tipo, una para cada cámara del cilindro, de forma que mediante un accionamiento eléctrico externo se pueda desplazar el vástago hacia delante, hacia atrás, o dejarlo en posición de reposo.

Fig. 2.19: Válvula *MOFH-3-1/8*

La conexión eléctrica de estas válvulas se efectuará mediante un cable *KMF-1-24-2,5-LED* compuesto interiormente de tres pequeños cables, uno amarillo y verde, que deberá unirse al terminal negativo de la fuente de alimentación; y dos cables negros, que han de conectarse indistintamente a 0 V y a 24 V

para activar la válvula. Estos últimos cables partirán de una entrada/salida digital y de una masa digital de la bornera y, mediante unos circuitos adaptadores de tensión, proporcionarán a la válvula los 24 V necesarios para su activación.

Es importante destacar que estas válvulas conmutan únicamente si se encuentran activas y existe una diferencia de presión entre el conducto de entrada y el de salida de al menos 1.5 bar.

Las soluciones a posibles fallos relacionados con estas válvulas se recogen en la figura 2.20.

Funcionamiento defectuoso	Causa posible	Solución
La válvula no conmuta	No se ha observado el margen de presión permisible	Verificar la presión ajustada en el control o antes de la válvula, especialmente si hay conectados muchos dispositivos consumidores.
	Fallo de conexión	Verificar las conexiones neumáticas y eléctricas, si procede.
	Válvula defectuosa	Devolver el producto a Festo para reparar.
Fuerte ruido en la conmutación	Elevado ruido del escape de aire	Utilizar silenciadores.

Fig. 2.20: Posibles fallos y soluciones de válvulas *MOFH*

2.9. Válvula reguladora de presión *LFR-D-MINI*.

Se utilizará una válvula reguladora *LFR-D-MINI* (figura 2.21) para compensar las fluctuaciones a la entrada de aire comprimido y ajustar su presión. Además, esta unidad filtra las partículas de suciedad y el condensado que pueda contener el aire.



Fig. 2.21: Válvula *LFR-D-MINI*

2.10. Fuente de alimentación.

Todos los componentes eléctricos mencionados en este apartado requieren una tensión de alimentación de 24 V D.C., excepto la tarjeta de adquisición de datos que se alimenta a través del puerto PCI del ordenador.

Realizando un cálculo aproximado del consumo de los convertidores, válvulas y sensor, se estima que la fuente ha de proporcionar una intensidad de al menos 3 A.

ELEMENTO	CONSUMO DE INTENSIDAD
Convertidor DADE-MVC-010	2 x 25 mA
Válvulas proporcionales MPYE 5-1/8-LF-010-B	2 x 833 mA
Solenoides MOFH-3-1/8	2 x 187 mA
Sensor SMAT-8E-S50-IU-M8	24 mA
TOTAL	2.109 A

La fuente utilizada en el presente proyecto proporciona 24 V y 5 A de corriente continua.



Fig. 2.22: Fuente de alimentación

2.11. Circuitos de potencia.

La tarjeta de adquisición de datos empleada dispone de varias salidas digitales que proporcionan una tensión entre 4.3 y 5.1 V. Estos valores son insuficientes para activar las válvulas digitales *MOFH* o las entradas de referencia y calibración de los convertidores *DADE*. Para resolver esta situación se construyeron dos circuitos adaptadores de tensión para las válvulas y un tercer circuito interruptor mediante accionamiento eléctrico.

Los circuitos adaptadores de tensión para las válvulas *MOFH* se encontraban en perfecto estado, por lo que han sido utilizados tal y como se implementaron en el proyecto “*Montaje, programación y puesta en marcha de un robot neumático de estructura paralela*”, Rosillo 2007.

Sin embargo, el interruptor mediante accionamiento eléctrico no se encontraba operativo, por lo que fue necesario crear uno nuevo. En este proyecto se optó por pedir un dispositivo convertidor de tensión continua, de forma que se pudiera transformar una señal de 5 V en otra de 24 V con un solo elemento y una alta eficiencia.

El dispositivo utilizado es el *DC/DC-Converter RN-0524S*, de 1.25 W de consumo de potencia y una eficiencia en torno al 80-85%.



Part Number		Input Voltage (VDC)	Output Voltage (VDC)	Output Current (mA)	Efficiency (%)	Max. Capacitive Load ⁽¹⁾
DIP6	(2kV)					
RN-xx24S	(H)	3.3, 5, 9, 12, 15, 24	24	52	80-85	220µF

Fig. 2.23: *DC/DC-Converter RN-0524S*

2.12. Otros elementos utilizados.

Para el conexionado neumático del robot se han utilizado otros elementos importantes, como tubos flexibles, distribuidores múltiples, rácores, silenciadores y tapones para las válvulas.



Fig. 2.24: Rácor, silenciador y distribuidor múltiple

CAPÍTULO 3:
SOFTWARE Y DESARROLLO DEL
PROGRAMA INFORMÁTICO.

3.1. Software utilizado.

En el desarrollo de este proyecto se ha hecho uso principalmente del programa *Matlab* y sus diferentes *toolboxes*.

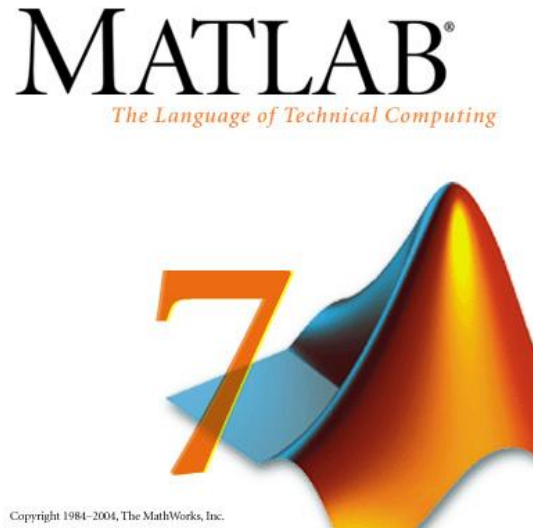


Fig. 3.1

3.1.1. Descripción y motivos de su uso.

Se trata de un programa para realizar cálculos numéricos con vectores, matrices, escalares, cadenas de caracteres y, en general, con estructuras de información complejas. Posee una gran variedad de herramientas para modelar sistemas de control, análisis, simulación y procesamiento de prototipos, lo que lo convierte en un sistema ideal para estudiar sistemas avanzados de control.

Matlab es un entorno de cálculo técnico de altas prestaciones para el cálculo numérico y visualización, que integra:

- Análisis numérico
- Cálculo matricial
- Procesamiento de señales
- Gráficos

Tiene un entorno fácil de usar, donde los problemas y soluciones son expresados como se escriben matemáticamente, sin la programación tradicional. *Matlab* es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensionamiento. Esto permite resolver muchos problemas numéricos en una fracción del tiempo que llevaría hacerlo en otros lenguajes de programación. *Matlab* ha evolucionado en los últimos años a partir de la colaboración de muchos usuarios. En entornos universitarios se ha convertido en la herramienta de enseñanza estándar para cursos de introducción en álgebra lineal aplicada, así como cursos avanzados en otras áreas. En la industria se utiliza para investigación y para resolver problemas prácticos de ingeniería y matemáticas, con gran énfasis en aplicaciones de control y procesamiento de señales. *Matlab* también proporciona una serie de soluciones específicas denominadas *toolboxes*. Éstas son muy importantes para la mayoría

de los usuarios y son conjuntos de funciones que extienden el entorno para resolver clases particulares de problemas como:

- Procesamiento de señales
- Diseño de sistemas de control
- Simulación de sistemas dinámicos
- Identificación de sistemas
- Redes neuronales y otros.

Una de las características más importantes de *Matlab* es su constante crecimiento, permitiendo al usuario crear sus propias aplicaciones. A modo de resumen, las prestaciones más destacadas de este programa son:

- Escritura del programa en lenguaje matemático.
- Implementación de las matrices como elemento básico del lenguaje, lo que permite una gran reducción de código.
- Implementación de aritmética compleja.
- Gran variedad de órdenes específicas, agrupadas en *toolboxes*.
- Posibilidad de ampliar y adaptar el lenguaje mediante ficheros script y funciones *.m*.

3.1.2. Herramientas utilizadas.

El código necesario para el funcionamiento del robot ha sido realizado en archivos *.m*. La herramienta *Simulink* ha servido de apoyo importante a la hora de realizar pruebas con las funciones de transferencia aproximadas asociadas al comportamiento de los cilindros.

Matlab dispone de multitud de *toolboxes* que facilitan la creación de las aplicaciones incluyendo numerosas funciones que no necesitan, por tanto, ser implementadas. Para el funcionamiento del robot se ha hecho uso fundamentalmente del *Data Acquisition Toolbox*, que contiene las funciones para la adquisición de datos de la tarjeta, es decir, para la lectura de entradas y salidas analógicas y digitales.

El motor de adquisición de datos de *Matlab* es un *MEX-file* o *dynamic-link library (DLL)* cuyas funciones son:

- Almacenar los objetos de los dispositivos y asociarles valores a las propiedades que controlan la aplicación de adquisición de datos.
- Controlar la sincronización de eventos.
- Controlar el almacenamiento de los datos adquiridos o puestos en cola.

Este motor y *Matlab* son asíncronos, por lo que se pueden usar simultáneamente con el consecuente ahorro de tiempo.

Otra herramienta destacada de *Matlab* es la función *ident*, que ha sido de gran ayuda para obtener funciones de transferencia aproximadas de los cilindros del sistema mediante la entrada como referencia de un vector de impulsos PRBS.

3.2. Desarrollo del programa informático.

Dentro del *Data Acquisition Toolbox* de *Matlab* hay numerosas funciones disponibles con las que enviar u obtener datos, además de las necesarias para crear objetos de entrada y salida analógicos o digitales.

A continuación se muestra un resumen de las funciones más destacadas que han servido para realizar las pruebas de este proyecto. Se indica si son funciones destinadas a objetos analógicos de entrada (AI) o de salida (AO), o a objetos de entrada y salida digital (DIO).

FUNCIÓN	DESCRIPCIÓN	AI	AO	DIO
FLUSHDATA	Borra los datos del motor de adquisición.	X		
GETSAMPLE	Adquiere inmediatamente una muestra	X		
GETDATA	Extrae datos, tiempo e información del evento.	X		
GETVALUE	Lee valores de las líneas.			X
PEEKDATA	Muestra el dato adquirido más reciente.	X		
PUTSAMPLE	Saca una muestra inmediatamente.		X	
PUTDATA	Pone datos a la cola del motor para una eventual salida.		X	
PUTVALUE	Escribe valores en las líneas.			X
ANALOGINPUT	Crea un objeto de entrada analógica.	X		
ANALOGOUTPUT	Crea un objeto de salida analógica.		X	
DIGITALIO	Crea un objeto de entrada/salida digital.			X
ADDCHANNEL	Añade canales de hardware para objetos de entrada o salida analógicos.	X	X	
ADDLINE	Añade líneas de hardware para objetos de i/o digital.			X
ISLOGGING	Determina si la entrada analógica está recibiendo datos (retorna un 1 en ese caso).	X		
ISSENDING	Determina si se están enviando datos por las líneas de salida analógicas (retorna un 1 en ese caso).		X	
ISRUNNING	Indica si está ejecutando un objeto (retorna un 1 en ese caso).	X	X	X
START	Comienza a ejecutarse el objeto.	X	X	X
STOP	Deja de ejecutarse el objeto.	X	X	X

Tabla 3.1: Funciones del *Data Acquisition Toolbox*

Estas funciones permiten transmitir información entre el PC y la tarjeta de adquisición de datos, y entre ésta y los componentes del robot.

Los programas realizados comprenden objetivos muy variados, como establecer la referencia o calibración de los cilindros, ejecutar el control de los dispositivos o integrar la cinemática del sistema. El objeto de este capítulo no es mostrar todo el código de los programas, que se hará en el anexo correspondiente, sino explicar el desarrollo de los mismos.

Es importante destacar que al crear objetos de entrada o salida, analógicos o digitales, se debe especificar el driver y el número de identificación de la tarjeta de adquisición de datos. Para la tarjeta de *National Instrument* empleada, estos valores son los siguientes:

Tarjeta de adquisición de datos <i>National Instrument</i>	
Driver	<i>nidaq</i>
nº de ID	1

Tabla 3.2: Características de la tarjeta de adquisición de datos utilizada

3.2.1. Programas para la puesta en marcha.

3.2.1.1. Recibir datos de sensores.

Al conectar nuevos sensores a las entradas analógicas del dispositivo de adquisición de datos conviene realizar una prueba para comprobar que los resultados se corresponden con lo que debería marcar el sensor. Si esto no es así, puede que aparezca una fuerte señal de ruido debido a una mala conexión del dispositivo.

Se debe indicar el canal en el cual se van a tomar las medidas, que depende del terminal escogido en la bornera. Del mismo modo, siempre es útil especificar los rangos en los que se quiere trabajar y las muestras por segundo que se quieren tomar, aunque en muchos casos se han dejado por defecto ya que los valores son adecuados.

El programa quedaría de la siguiente forma:

```
%Crea objeto de entrada analógica
ai = analoginput('nidaq',1)

%Crea canal para el objeto
chans_i = addchannel(ai,0)

%Establece la frecuencia de muestreo y el número de muestras por disparo
set(ai, 'SampleRate', 30)
set(ai, 'SamplesPerTrigger', 100)

%Establece los rangos de trabajo.
set(chans_i, 'SensorRange', [0 10])
set(chans_i, 'InputRange', [0 10])
set(chans_i, 'UnitsRange', [0 10])

%Ejecución del objeto
start(ai);

%Recibe los datos del sensor y los muestra en una gráfica
h=getdata(ai);
plot(h);

%Detiene la ejecución del objeto y lo elimina
stop(ai);
delete(ai);
```

3.2.1.2. Establecimiento del punto de referencia.

Debido a la necesidad de establecer la referencia de los cilindros cada vez que se conecta el sistema, se creó este programa para automatizar este proceso.

El programa crea un objeto de salida digital y envía 5 V a través de él para que, a través de un convertidor de tensión continua, le lleguen a los convertidores los 24 V necesarios para su referencia. Esta tensión ha de ser mantenida durante más de dos segundos, por lo que se introdujo un contador de una duración superior.


```

%Crea objeto de i/o digital y línea.
dio = digitalio('nidaq',1);
addline(dio,3,0,'out');

%Saca 5V al cable gris para establecer punto referencia
putvalue(dio,1)
tic
for i=1:50000^2    %Contador de duración 6 seg. aprox.
end
%Saca 0V al cable gris para finalizar referencia
putvalue(dio,0)
toc

delete(dio)
clear dio

```

Este programa se puede usar también para establecer el punto de calibración simplemente intercambiando el cable gris por el cable rosa a la salida del terminal *DIO3* de la bornera.

3.2.2. Control de la posición de los cilindros DNCL.

Para llevar a cabo el desplazamiento de los cilindros y el control de su posición se realizaron tres funciones. Estos *scripts* han sufrido diversas modificaciones a lo largo del desarrollo del proyecto ya que, por ejemplo, al principio controlaban la posición de un solo cilindro y después evolucionaron para incluir el control de los dos cilindros.

A continuación se explica el objetivo de cada función y sus aspectos más destacados.

3.2.2.1. Función *PID*.

Los cilindros *DNCL-32-300-P-A* son accionados mediante válvulas proporcionales *MPYE-5-1/8-LF-010-B* que controlan tanto el sentido de avance como el caudal de aire que introducen en cada momento.

La función es llamada mediante el comando `[ai,ao]=PID`.

```
function [ai,ao] = PID()
```

Las variables declaradas aquí han de ser globales para que el resto de funciones puedan recurrir a ellas. Se crearán dos variables de señal de referencia, de salida y de señal de control (una para cada cilindro), así como dos variables auxiliares: *INTEGRAL* y *N_MUESTRA*, que se usarán en la siguiente función encargada del control propiamente dicho.

```

global S_REFERENCIA1;
global S_REFERENCIA2;
global SALIDA1;
global SALIDA2;
global INTEGRAL;
global N_MUESTRA;
global S_CONTROL1;
global S_CONTROL2;

```

Después de inicializar las variables para acelerar el proceso de cálculo, se crearán los objetos de entrada y salida analógica, con sus respectivos canales, y se procederá a ajustar los rangos de trabajo de los sensores y actuadores y la frecuencia de muestreo.

```
%crea los objetos ai y ao y asigna los canales
ao = analogoutput('nidaq',1);
ai = analoginput('nidaq',1);
chans_o = addchannel(ao,0:1);
chans_i = addchannel(ai,0:1);

%Ajusta los rangos de las entradas a los de los sensores
set(chans_i, 'SensorRange', [0 10]);
set(chans_i, 'InputRange', [0 10]);
set(chans_i, 'UnitsRange', [0 10]);

%ajusta los rangos de salida a los de los actuadores
set(chans_o, 'OutputRange', [0 10]);
set(chans_o, 'UnitsRange', [0 10]);

Frecuencia de muestreo (Hertzios)
set(ai, 'SampleRate', 30);
ai.SamplesAcquiredFcnCount = 1;
```

La función que implementa el algoritmo de control se define del siguiente modo:

```
ai.SamplesAcquiredFcn = {@callback_PID, ao};
```

3.2.2.2. Función *CALLBACK_PID*.

Esta función es llamada por la anterior para cada muestra adquirida, contiene los parámetros de los controladores, la eliminación de zonas muertas, la saturación y la eliminación del fenómeno *windup*.

Los parámetros de los controladores se especifican del siguiente modo:

```
Kp(1)=2;
Ki(1)=0.001;
Kd(1)=0.02;

Kp(2)=2.5;
Ki(2)=0.001;
Kd(2)=0.025;
```

A continuación se implementan los cilindros. Para ello se almacenan en la variable *SALIDA* los datos tomados por los sensores gracias al comando *getsample*. Dicho valor se almacena también en una variable indicadora de la posición. Además se debe guardar el error de cada muestra y un parámetro de error acumulado.

```
%Posicion
Y_k(1) = sal(1);
%Error
E_k(1) = REFERENCIA1(N_MUESTRA)-Y_k(1);
%Integral
INTEGRAL(1) = INTEGRAL(1) + E_k(1);
```

- **Zona muerta de los actuadores.**

La zona muerta de los actuadores es el mínimo cambio que debe producirse en la señal de control para que el cilindro se mueva. Esto provoca que, para una referencia fija dada, la válvula no cierre correctamente y el cilindro siga desplazándose, lo que provoca una reacción inmediata en la señal de control que vuelve a activar la válvula (ver figura 3.2).

Para solucionar este problema, a la señal de control se le indica que, si la posición del cilindro es igual a la referencia más menos el error de cero, envíe un valor de tensión previamente calculado que mantenga la válvula completamente cerrada. De este modo, si la posición del cilindro se acerca lo suficiente a la referencia, la señal de control se mantendrá constante y el vástago no se alejará de ese valor más que lo debido a perturbaciones externas (ver figura 3.3).

En caso contrario, la señal de control será la suma de las acciones proporcional, integral y derivativa del controlador.

```
if REFERENCIAL(N_MUESTRA) - 0.12 < Y_k(1) && REFERENCIAL(N_MUESTRA) + 0.12 > Y_k(1)
    U(1) = -0.1;
else
    U(1) = Kp(1) * E_k(1) + Ki(1) * INTEGRAL(1) + Kd(1) * (E_k(1) -
E_k1(1)) / (1/30);
end
```

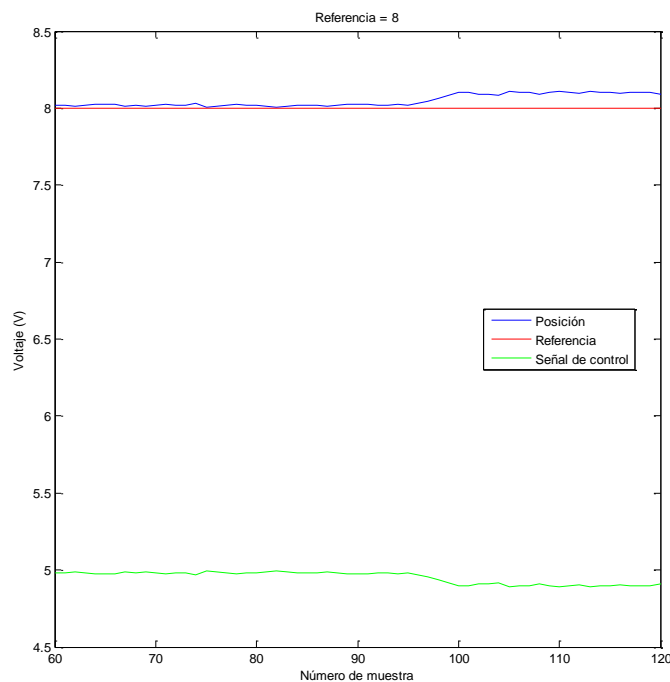


Fig. 3.2: Efectos del cierre incorrecto de válvulas.

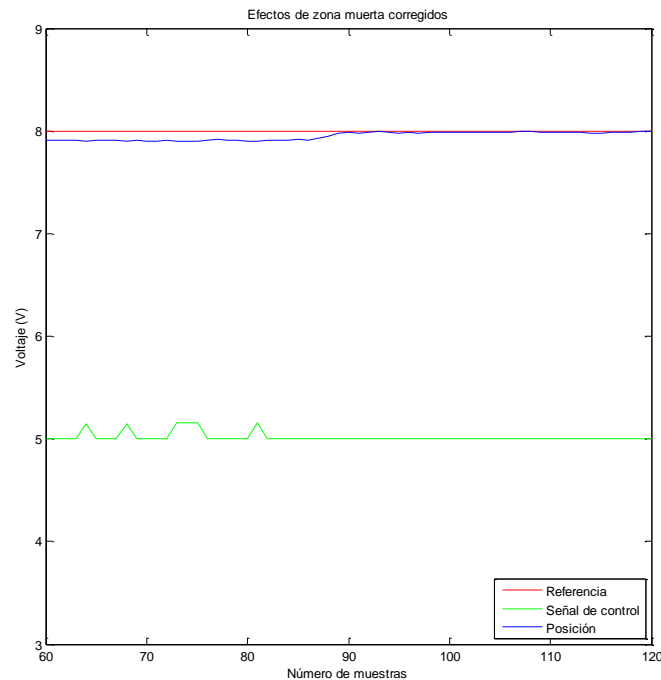


Fig. 3.3: Cierre completo de válvula.

La señal de control implementada oscila en torno al cero. Para valores mayores a 5 y menores que -5 la válvula vibra y se vuelve incontrolable por lo que se satura en esos valores y, para adecuarla a los rangos de los sensores y actuadores, hay que sumarle 5 unidades.

- **Fenómeno “windup”.**

Uno de los problemas que pueden dar lugar a sobrepicos o inestabilidad es el fenómeno del *windup* que se presenta en sistemas con acción de control integral en los que existe una saturación en el actuador. Cuando la señal de control está saturada, la magnitud de salida del integrador continuará en aumento (a menos que la salida del integrador también alcance un límite) sin ningún efecto sobre la respuesta del sistema.

Si se supone que el error grande es un fenómeno transitorio, en algún instante éste caerá a cero y el signo cambiará. Sin embargo, aparecerá un retraso innecesario en la respuesta al cambio mientras el integrados retorna a un valor que restaurará la operación lineal. En otras palabras, se debe integrar una señal de error de signo opuesto durante un periodo de tiempo suficiente para devolver la salida del integrador al rango de operación normal.

Para evitar este fenómeno se puede diseñar un circuito de forma que active un lazo de realimentación negativo en torno al integrador durante el tiempo que dura la saturación, conocido como *anti-windup*. En control digital basta con dejar de sumar el error a la variable integral, del siguiente modo:

```
%Si la señal esta saturada evitamos que INTEGRAL aumente su valor
if U(1)>=6 || U(1)<=4
INTEGRAL(1) = INTEGRAL(1) - E_k(1);
end
```

Para terminar la implementación del cilindro introducimos una saturación conveniente de la señal de control para regular la velocidad de avance del vástago.

```
if U(1)>5.7
U(1)=5.7;
elseif U(1)<4.1
U(1)=4.1;
end
```

Por último se almacena la señal de control U en la variable $S_CONTROL$, se sacan los valores a los actuadores mediante el comando *putsample* y se actualizan el error y el número de muestra.

```
S_CONTROL1(1,N_MUESTRA)= U(1);
S_CONTROL2(1,N_MUESTRA)= U(2);

putsample(ao, [U(1) U(2)]);

%Actualización del error en el intervalo anterior
E_k1 = E_k;

%Pasamos a la muestra siguiente
N_MUESTRA = N_MUESTRA + 1;
```

La implementación del cilindro horizontal *DNC* difiere de la de los otros dos cilindros, ya que no requiere de ningún tipo de control asociado y resulta mucho más sencilla.

Para realizar el seguimiento de trayectorias se procuró que el cilindro actuador se desplazara hacia delante al comenzar el proceso y hacia atrás una vez que finalizara el recorrido de los cilindros anteriores. Las válvulas *MOFH* se activarán en la primera muestra para mover el vástago, y para retraerlo al llamar a la función *finaliza.m*.

```
if REFERENCIA3(N_MUESTRA)==1
A = 0;
B = 1;
else
end

putvalue(dio, [A B]);
```

Con esto queda completo el programa necesario para controlar la posición de los cilindros. No obstante cuando se propuso desarrollar el seguimiento de trayectorias, se implementó en este mismo script la cinemática del robot: la inversa para indicar la referencia a partir de coordenadas, y la directa para representar el punto del actuador.

Además se le añadieron otras mejoras como la introducción de vectores de almacenamiento de coordenadas que, según el periodo de muestreo indicado proporcionarían al sistema una velocidad de escritura determinada, y la inclusión de la función *finaliza* para que el programa termine automáticamente al llegar a la última coordenada especificada.

3.2.2.3. Función *FINALIZA*.

Esta función debe ser llamada cuando se desea detener la ejecución de los programas anteriores, en cualquier momento del proceso.

- Devuelve los cilindros a su posición inicial.

```
putsample(ao,[4,4]);
putvalue(dio,[1 0]);
```

- Detiene y elimina los objetos analógicos y digitales.

```
stop(ai);
stop(ao);
delete(ai);
delete(ao);
stop(dio);
delete(dio);
```

- Muestra en pantalla las gráficas de la posición, referencia y señal de control de cada cilindro.

```
figure()
plot(S_REFERENCIAL,'r');
hold on;
plot(S_CONTROL1,'g');
hold on;
plot(SALIDA1);
hold on;
title('CILINDRO 1')
legend('Referencia','Señal de control','Posición')
xlabel('Número de muestras')
ylabel('Voltaje (V)')
```

- Para el seguimiento de trayectorias esta función también muestra la gráfica de la posición del actuador y la figura ideal que se pretendía representar.

```
%representamos el punto de actuacion del actuador
figure();
axis([-20,80,0,80])
xlabel('distancia horizontal (cm)');
ylabel('distancia vertical (cm)');
plot(xp,yp,'r')
axis equal
hold on

%Representación de la figura geométrica teórica
plot(X,Y,'b');
```

Su sintaxis es *finaliza(ai,ao)*.

3.2.3. Cinemática del robot.

Para poder realizar seguimientos de trayectorias con el robot es imprescindible implementar su cinemática en código *Matlab*. A continuación se describen los procedimientos para obtener la posición del extremo final actuador conociendo los valores de las coordenadas articulares (cinemática directa) y viceversa (cinemática inversa).

Los parámetros que caracterizan al sistema son los siguientes:

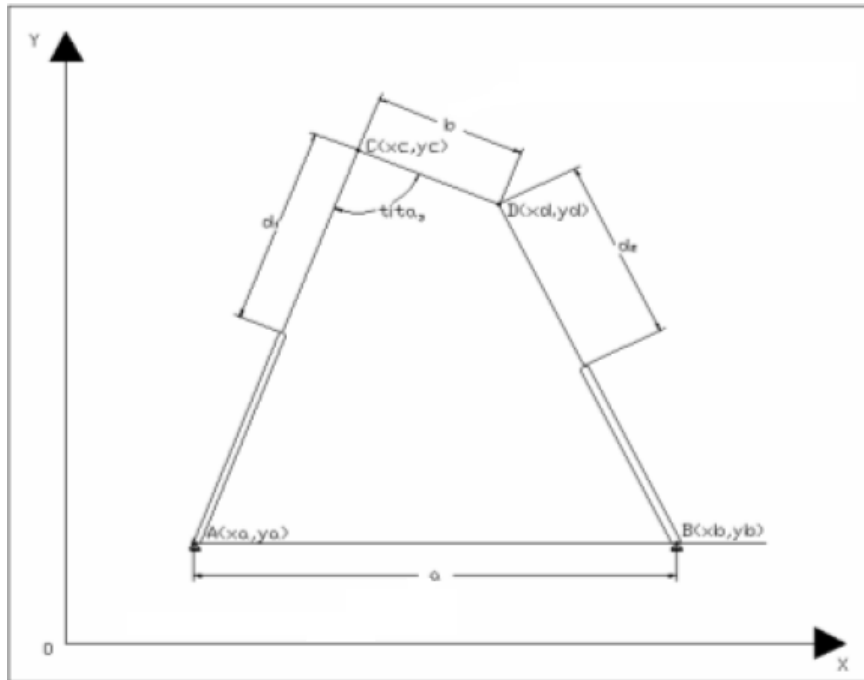


Fig. 3.4: Parámetros de la estructura robótica.

```
a=60; %distancia entre apoyos (cm)
b=20; %longitud del eslabón de unión (cm)
factor=0.5; %factor numérico que sitúa el efector sobre el eslabón
sit=b*(factor); %situación del efector sobre el eslabón de unión (cm)
tita3=2*pi/3; %ángulo entre el cilindro 1 y el eslabón unión (radianes)
l=50; %longitud del cilindro (cm)
%coordenadas de las articulaciones fijas del robot (cm)
xa=0;
ya=0;
xb=a;
yb=0;
```

3.2.3.1. Cinemática directa, *CDIR*.

En esta función se realiza el proceso de obtener la posición cartesiana (xp, yp) del efector final a partir de los valores conocidos de las coordenadas articulares *d1* y *d2*.

Su sintaxis es *cdir(d1,d2)*.

```
function [xp,yp]=cdir(q1,q2);
parametros
upa=sqrt((l+q1)^2+(b^2)-(2*(l+q1)*b*cos(tita3)));
upb=(l+q2);
tita2=pi-(tita3);

%2*(xb-xa)*x+2*(yb-ya)*y+R-S=0;
R=(xa^2)+(ya^2)-upa^2;
S=(xb^2)+(yb^2)-upb^2;
M=(ya-yb)/(xb-xa);
N=(1/2)*((S-R)/(xb-xa));

%p*(y^2) + q*y + r = 0
p=(M^2)+1;
q=2*((M*N)-(M*xa)-ya);
r=(N^2)-(2*N*xa)+R;
z=(q^2-4*p*r);
```

A continuación se establece la condición que deben cumplir las ecuaciones cuadráticas para trabajar en el dominio del campo real.

$$z = \sqrt{b^2 - 4ac} > 0$$

Se calcula además la situación de los puntos *C* y *D*, extremos de la pieza de unión entre los vástagos de los cilindros.

```
if z>=0
yd=(-q+sqrt(q^2-4*p*r))/(2*p);
xd=M*yd+N;

%calculo del angulo tita2
d=sqrt((xd^2)+(yd^2));
alfa=atan(yd/xd);
beta=acos((b^2-(l+q1)^2-d^2)/(-2*d*(l+q1)));
tita2=alfa+beta;
xc=(l+q1)*cos(tita2);
yc=(l+q1)*sin(tita2);
```

La otra solución de la ecuación, $y_d = (-q - \sqrt{q^2 - 4pr}) / (2p)$, da el eje negativo de la coordenada *y*, por lo que es físicamente imposible.

Por último se calcula la posición del efector final.

```
if yc>=0
%calculo de la posicion del actuador
fi=tita2+tita3-pi;
xp=xc+(sit)*cos(fi);
yp=yc+(sit)*sin(fi);
```


3.2.3.2. Cinemática inversa, *CINV*.

En el problema inverso de la cinemática se determinan las variables articulares $d1$ y $d2$, conocida la posición del actuador. Este programa resulta fundamental para el seguimiento de trayectorias ya que transforma las coordenadas de las figuras a dibujar en longitudes de los vástagos y, por lo tanto, en valores de tensión que introducir a cada válvula dada la relación $10V = 30 \text{ cm}$.

```
function [q1,q2]=cinv(xp,yp);
format long
parametros

q1=0;
q2=0;

%parametros geometricos necesarios
alfa=atan(yp/xp);
d=sqrt((yp^2)+(xp^2));
beta=asin((sit*sin(tita3))/d);
tita2=alfa+beta;
fi=tita2+tita3-pi;

%calculo del punto C para calcular d1
xc=px-sit*cos(fi);
yc=py-sit*sin(fi);
m=sqrt((yc^2)+(xc^2));
d1=m-l;

%calculo del punto D para calcular d2
xd=px+sit*cos(fi);
yd=py+sit*sin(fi);
n=sqrt((yd^2)+(a-xd)^2);
d2=n-l;
q1=d1;
q2=d2;
```


CAPÍTULO 4:

FUNDAMENTOS DE CONTROL Y PUESTA EN MARCHA DEL SISTEMA.

4.1. Introducción.

En este apartado se describirá la puesta en marcha del sistema, desde los pasos a seguir para su correcto funcionamiento hasta los mecanismos de control que se han utilizado.

Se hará especial hincapié en la elección de la presión de aire de suministro y el periodo de muestreo, cuestiones fundamentales para que el robot responda de una forma u otra.

Se hará una introducción a los mecanismos de control empleados en el presente proyecto, explicando de forma teórica los aspectos más importantes para el desarrollo de los siguientes apartados.

4.2. Puesta en marcha.

Para evitar movimientos bruscos de los cilindros y cualquier posible malfuncionamiento de los aparatos es conveniente seguir el siguiente orden:

1. Encender el PC, de forma que se alimente la tarjeta de adquisición de datos.
2. Encender la fuente de alimentación que presta servicio a las válvulas y demás componentes.
3. Abrir la válvula de aire comprimido.

De esta forma el sistema se encuentra operativo en el momento de inyectar el aire, por lo que se minimizan las posibilidades de movimientos bruscos de los cilindros.

4.2.1. Referencia y calibración de los cilindros.

Se establecerá la referencia de los cilindros, mediante el script apropiado para ello, preferentemente antes de abrir el aire comprimido para evitar posibles movimientos bruscos al conectar el aire o, al menos, asegurándose de que ambos vástagos se encuentran en su posición más retraída. Una vez realizado este proceso la gráfica correspondiente a los cilindros en su posición inicial es la siguiente:

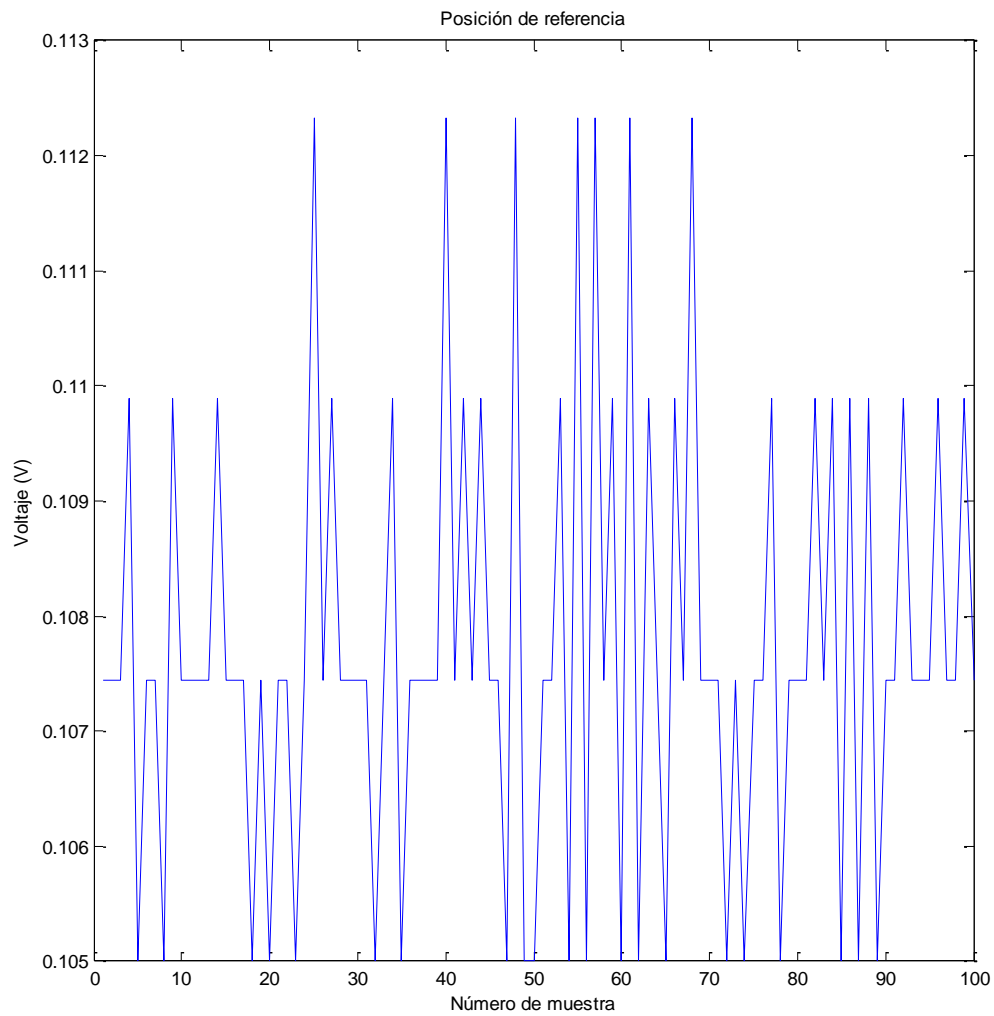


Fig. 4.1. Cilindro en posición de referencia

Si es la primera vez que se conecta el sistema, o si la memoria ha sido restablecida, además de la referencia será necesario establecer la calibración de los convertidores *DADE* para la posición final de los cilindros. Esto se hará según lo explicado en el apartado de componentes. No obstante, se puede realizar mediante el script de referencia simplemente intercambiando el cable de referencia por el de calibración (ver apartado componentes: convertidores *DADE*) a la salida del convertidor de tensión DC/DC. Si el sistema se ha calibrado correctamente, la gráfica correspondiente a cualquiera de los cilindros en su posición más extendida tiene la siguiente forma:

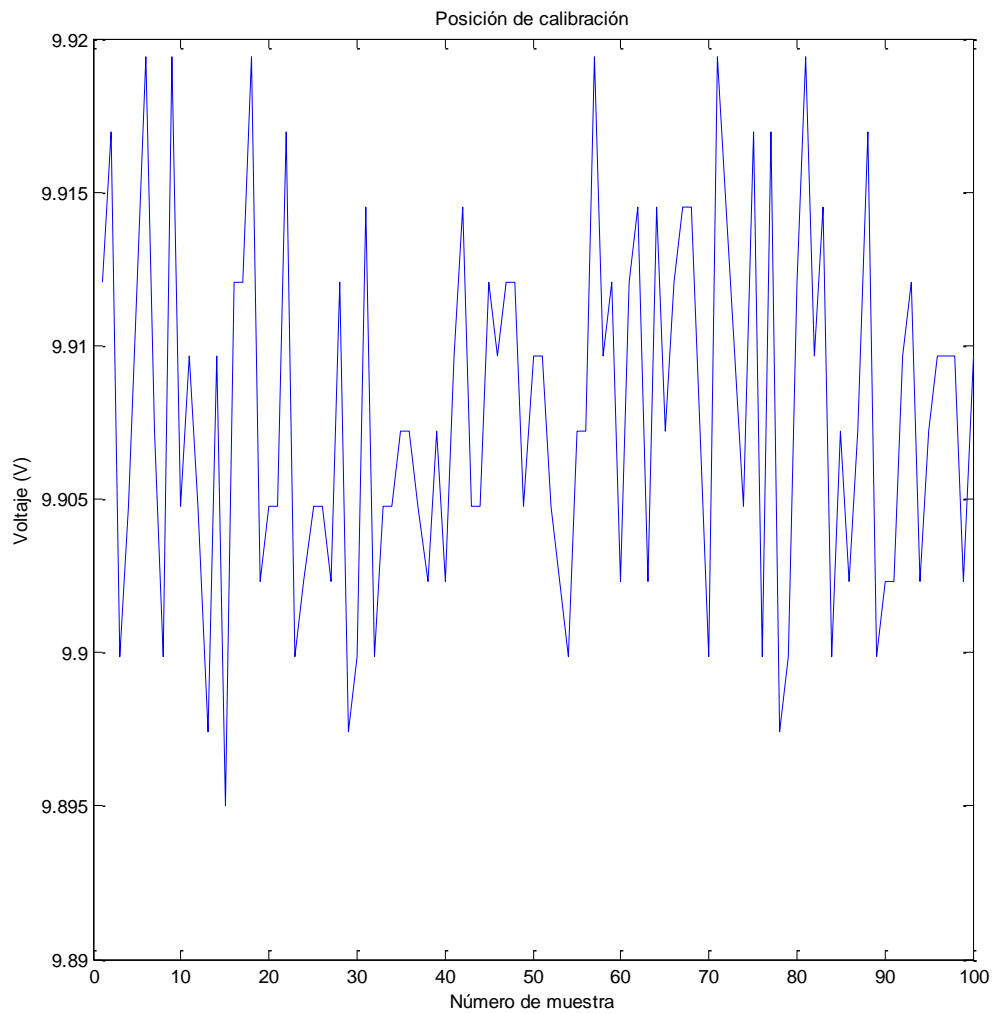


Fig. 4.2. Cilindros en posición de calibración

Observamos en estas gráficas un error de cero de aproximadamente 0.11 V, que será tenido en cuenta a la hora de implementar el código de control y evitar los problemas de zona muerta de las válvulas proporcionales.

Con el sistema referenciado y calibrado, el desplazamiento manual de uno de los cilindros desde su posición inicial hasta cerca del final y de nuevo hasta el inicio ofrece una salida de la siguiente forma:

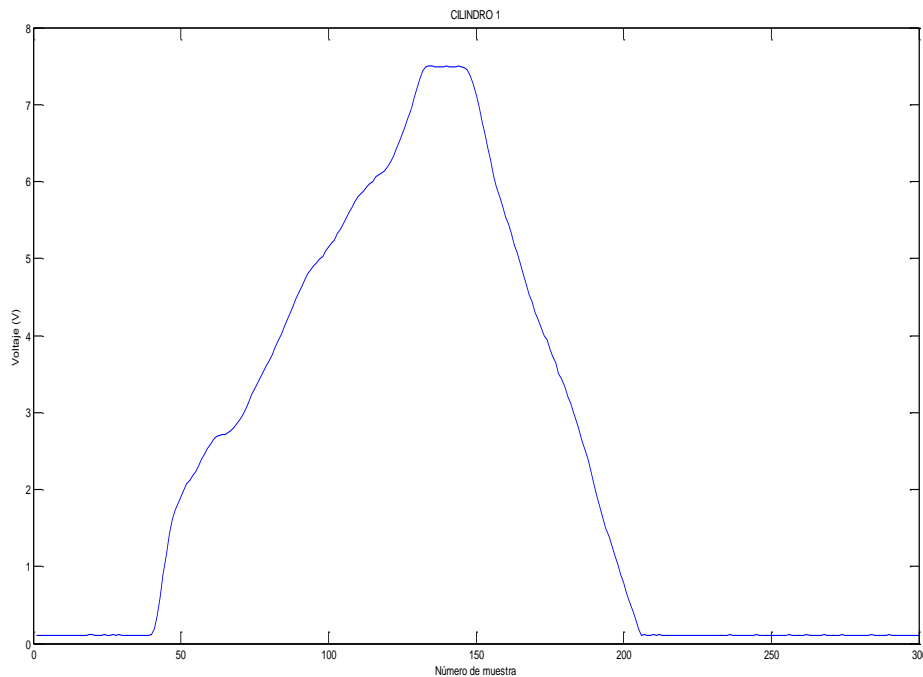


Fig. 4.3. Desplazamiento manual de un cilindro

4.2.2. Problemas tras el encendido.

Una vez que se conecta la fuente de alimentación se escucha un ruido procedente de las válvulas proporcionales, una vibración que desaparece en cuanto éstas reciben algún valor de tensión. Al alimentar estas válvulas, por el canal de entrada reciben un valor de tensión igual a 0, de modo que la válvula vibra. Es necesario introducir un valor de tensión entre 0,2 y 9,8 para evitar la saturación y eliminar el ruido.

Además, se observa una caída de presión en el regulador *LFR* debida a una pérdida por las válvulas *MOFH* que, de igual modo que las proporcionales, no tienen asignado un valor de tensión por el canal de entrada, con lo que no retienen el aire y este se escapa por sus aberturas con silenciadores. Este desajuste se corrige automáticamente al activar dichas válvulas, con lo que el sistema quedaría preparado para su uso.

4.3. Elección de la frecuencia de muestreo.

Uno de los primeros parámetros a establecer es el periodo de muestreo de las señales adquiridas, ya que su valor será necesario incluirlo en el código de desplazamiento de los cilindros. Se llevaron a cabo diversas pruebas de la respuesta del sistema para diferentes valores de frecuencia de muestreo.

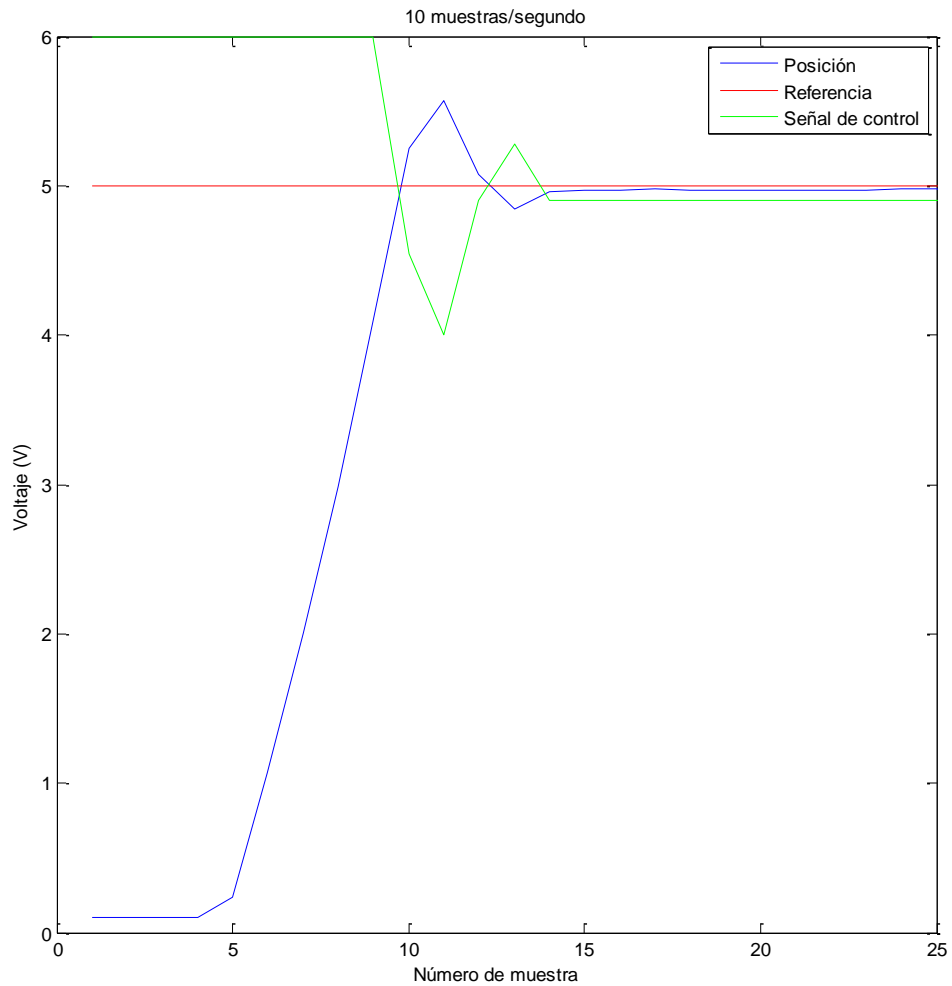


Fig. 4.4. Frecuencia de muestreo a 10 muestras por segundo

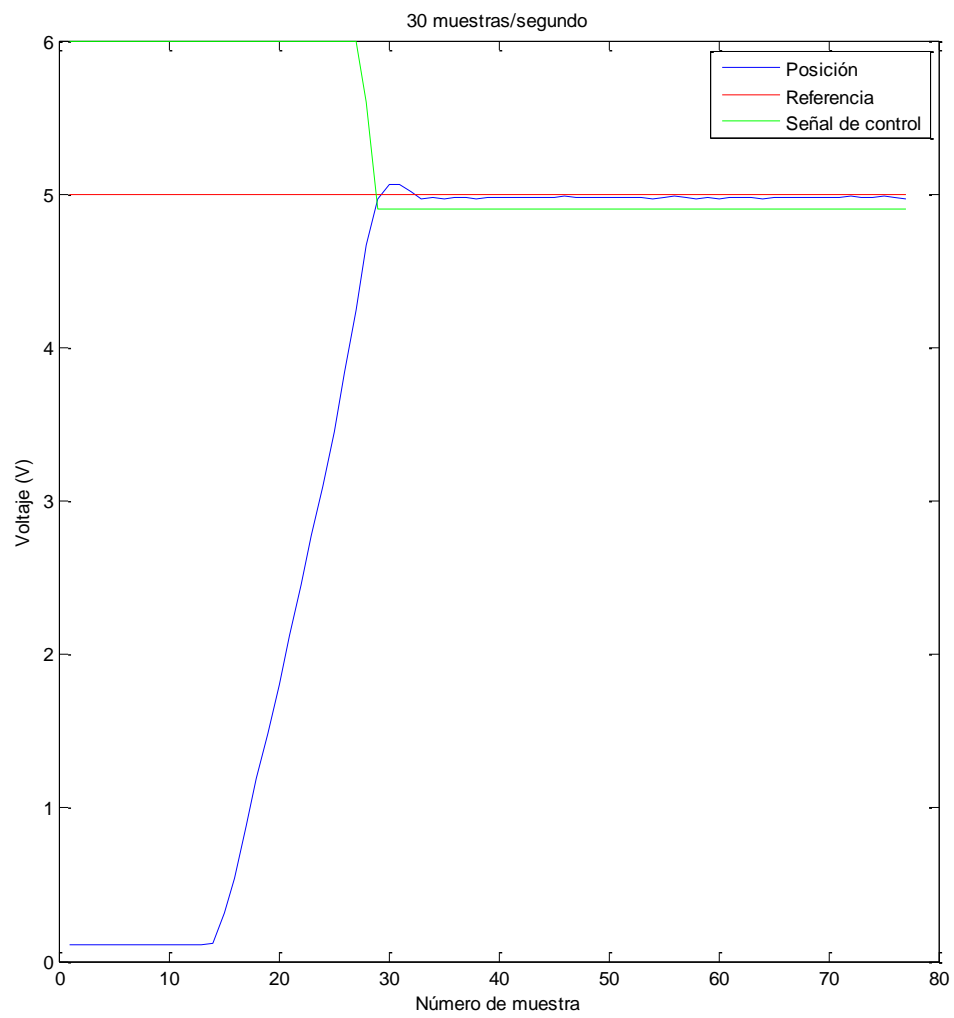


Fig. 4.5. Frecuencia de muestreo a 30 muestras por segundo

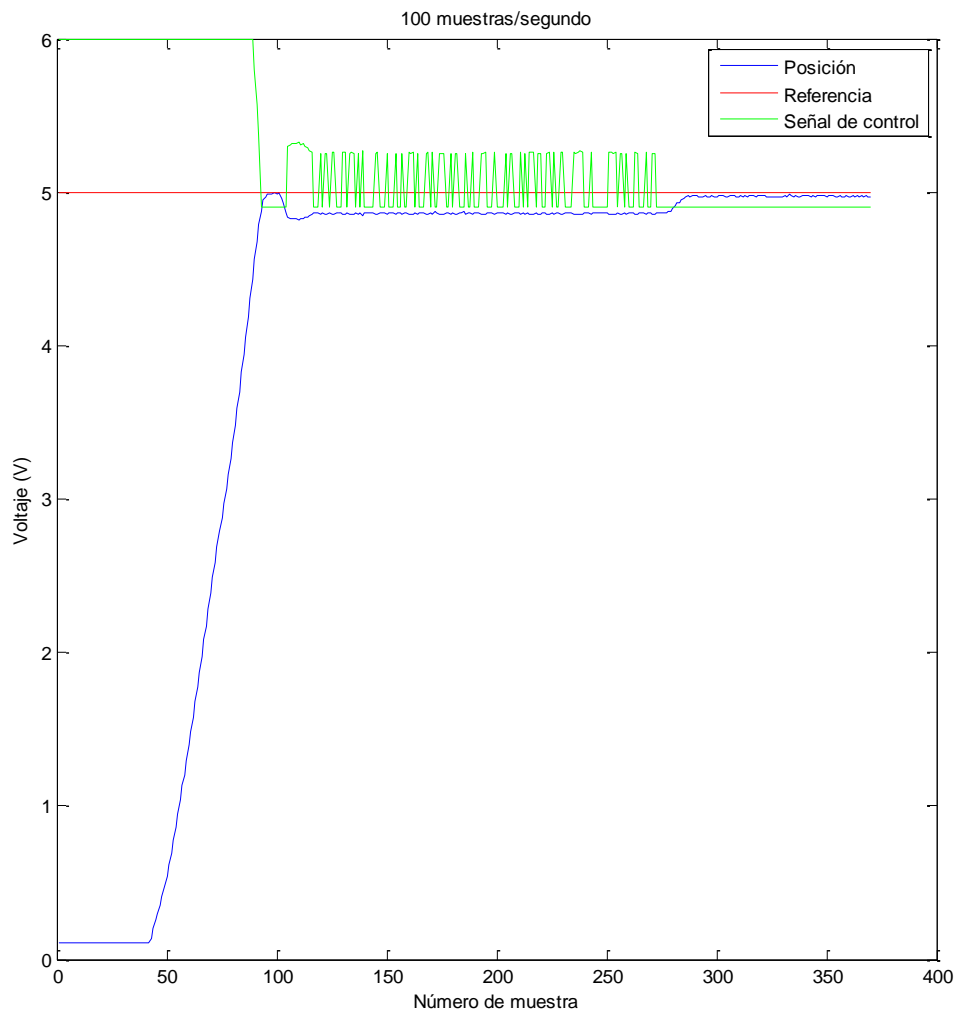


Fig. 4.6. Frecuencia de muestreo a 100 muestras por segundo

De las pruebas realizadas se observa que no hay una diferencia importante entre unos y otros periodos de muestreo aunque, si bien 10 muestras por segundo no son suficientes para controlar la posición del cilindro, con 100 muestras por segundo la señal de control no se mantiene constante y la válvula vibra.

Por tanto, de ahora en adelante se trabajará con una frecuencia de muestreo de 30 muestras por segundo.

4.4. Elección de la presión de suministro de aire comprimido.

Las variaciones en la presión de suministro afectan en gran medida a la velocidad de desplazamiento de los cilindros y, por tanto, a su estabilidad.

Se comenzó trabajando a una presión de 3 bar, con la que el sistema respondía de forma adecuada a grandes cambios en la referencia. Sin embargo, conforme se trataba de afinar el control y las diferencias entre valores de referencia se reducían, esta presión no era suficiente para obtener una respuesta suave de los cilindros, por lo que se aumentó hasta 5 bar.

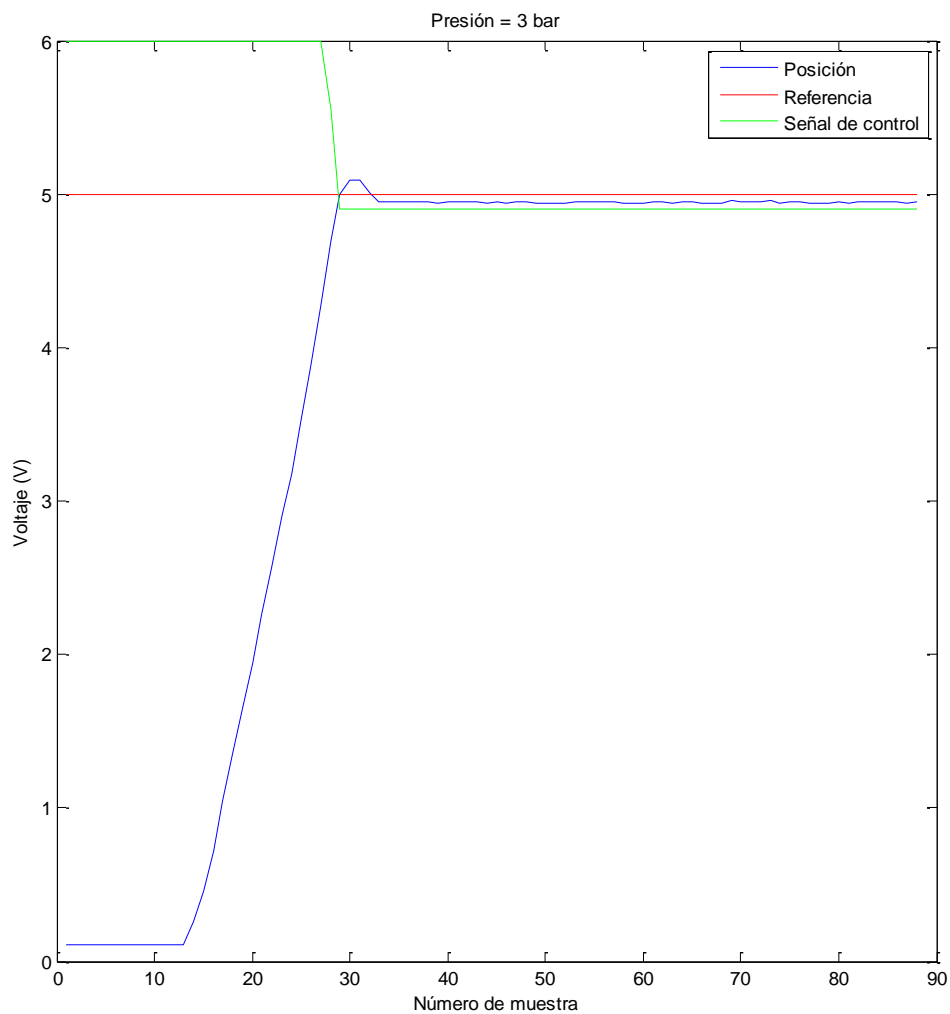


Fig. 4.7. Presión de suministro de aire comprimido a 3 bar

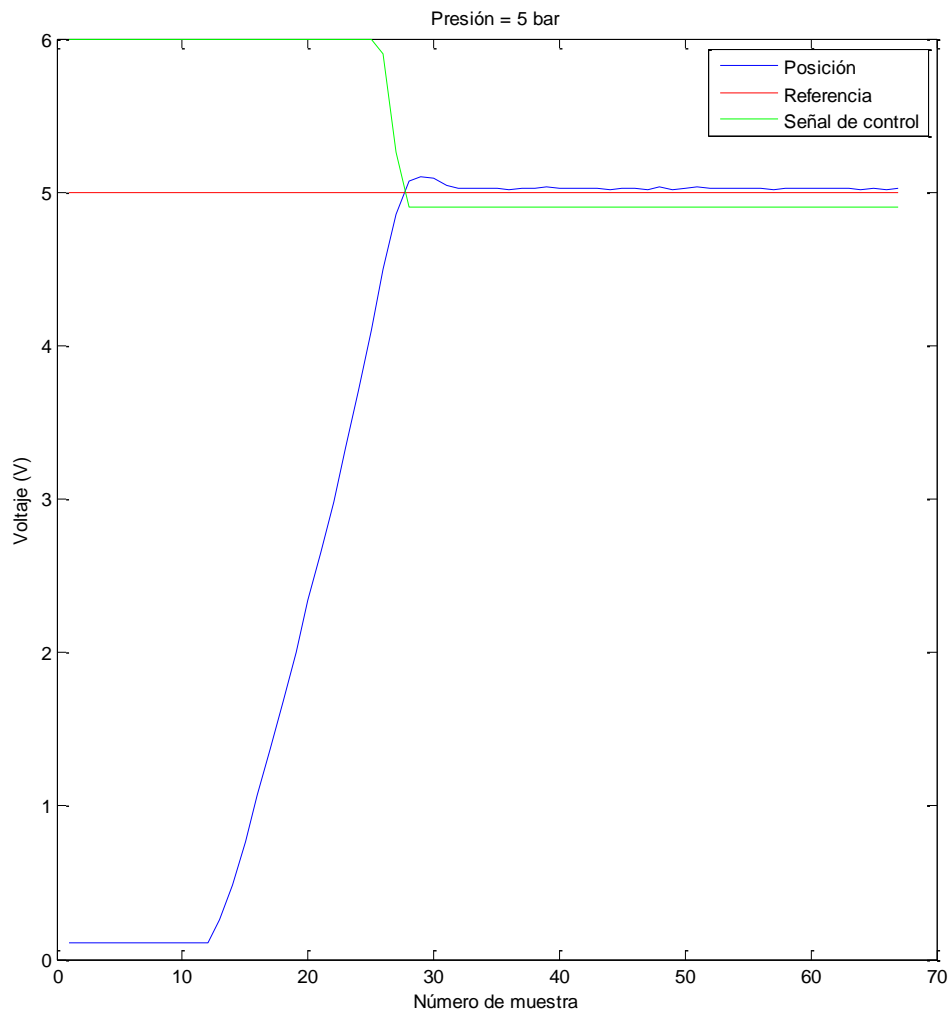


Fig. 4.8. Presión de suministro de aire comprimido a 5 bar

Las gráficas del sistema para una presión u otra apenas sufren modificaciones en cuanto a error de posición o tiempo en alcanzar la referencia. No obstante, con una presión de 5 bar se aprecia un movimiento más fluido de los cilindros lo que resulta absolutamente imprescindible para ejecutar un control de posición lo suficientemente preciso.

4.5. Introducción al control mediante PID.

En los experimentos llevados a cabo en este proyecto la variable a controlar es la posición del vástago del cilindro respecto a una señal de referencia introducida, estando ambos valores en escala de 0 a 10 voltios. El recorrido de los cilindros *DNCI* es de 300 mm por lo que un voltio de diferencia entre la referencia y la señal medida implicará un error de posición de 30 mm.

Los controladores PID son los más utilizados en toda clase de industrias, y ofrecen unas prestaciones muy aptas para la mayoría de sistemas de control. En el caso de desconocer el modelo matemático de la planta no es posible el uso de métodos de diseño de controladores analíticos, por lo que la sintonía experimental de controladores PID resulta de mucha utilidad.

4.5.1. Acciones básicas del control PID.

El control PID se puede esquematizar de la siguiente forma, siendo $G_p(s)$ el modelo matemático de la planta a controlar, y $G_c(s)$ la función de transferencia del controlador PID, dividido en sus tres acciones básicas: proporcional (K_p), integral (K_i) y derivativa (K_d).

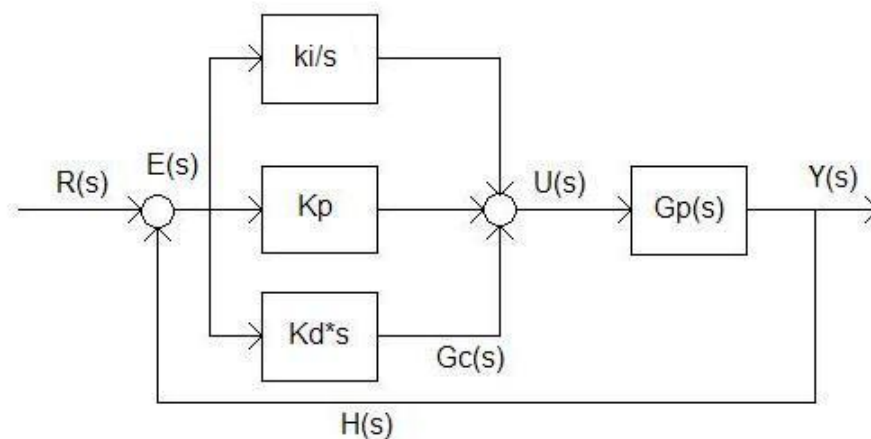


Fig. 4.9. Esquema de funcionamiento de un controlador PID

La función de transferencia de un controlador PID tiene la siguiente forma:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right)$$

siendo $K_i = \frac{K_p}{T_i}$ y $K_d = K_p \cdot T_d$

En función de estos parámetros (K_p , K_i y K_d) se pueden establecer las acciones básicas de control:

Acción de control	Tiempo de subida	Sobrepico	Tiempo de establecimiento	Error estacionario
K_p aumenta	Disminuye	Aumenta	Cambia poco	Disminuye
K_i aumenta	Disminuye	Aumenta	Aumenta	Eliminado
K_d aumenta	Cambia poco	Disminuye	Disminuye	Cambia poco

Estas reglas, aunque no son una ciencia exacta, han de ser tenidas en cuenta para la sintonización experimental de los controladores.

4.5.2. Sintonía de controladores PID.

Para sintonizar un controlador, el primer paso es escoger el tipo adecuado, proporcional (P), proporcional-integral (PI), proporcional-derivativo (PD), o PID. Para ello se requiere un mínimo conocimiento del efecto de las tres acciones y, a ser posible, tener experiencia sobre el proceso a controlar. A continuación se deben ajustar los parámetros para que la respuesta del sistema siga unas especificaciones determinadas.

El ajuste de parámetros PID es frecuente en muchos procesos industriales, tanto en operaciones de puesta en marcha como tras la detección de cambios sustanciales en el comportamiento de la planta. En caso de que sea posible obtener un modelo matemático de ella, se pueden aplicar diversas técnicas de diseño analítico de controladores con el fin de determinar los parámetros del mismo que mejor se ajustan a las características del proceso, ya sea en situación transitoria o estacionaria. Sin embargo, en multitud de ocasiones la planta es demasiado compleja para extraer un modelo matemático fiable, por lo que se requiere el empleo de técnicas experimentales. Estas son:

- Ensayos de prueba y error.
- Sintonía experimental en lazo abierto.
- Sintonía experimental en lazo cerrado.
- Sintonía experimental con especificaciones frecuenciales.

Las especificaciones de operación de la planta están sujetas a diversas limitaciones. Por ejemplo, con un controlador proporcional no se puede conseguir una máxima precisión y estabilidad, pues una mejora en una de ellas origina un perjuicio en la otra. Por este motivo, una sintonía PID experimental que implique los tres tipos de acciones de control (proporcional, integral y derivativa) es la opción que proporcionará mejores resultados.

CAPÍTULO 5:
**ANÁLISIS DEL MOVIMIENTO
DESACOPLADO DE LOS CILINDROS.**

5.1. Introducción.

En este apartado se describe el procedimiento seguido para establecer unos parámetros adecuados del controlador PID.

En primer lugar se llevaron a cabo unos primeros desplazamientos de los cilindros para estudiar su comportamiento en distintas condiciones. Tras estas pruebas se llegó a la conclusión de que era necesario implementar dos controladores, uno para cada cilindro.

El sistema, por tanto, es analizado como un sistema SISO (single input-single output), que consta de una única entrada (tensión en la válvula de un cilindro) y una salida (posición de dicho cilindro).

Se realizaron numerosas pruebas tanto en lazo abierto como en lazo cerrado para extraer una función de transferencia más o menos fiable para cada cilindro *DNCL*. Mediante simulaciones en *Simulink* se comprobaron las diferencias en el comportamiento de ambos cilindros, lo que reforzó la decisión de establecer dos controladores independientes, aunque no muy diferentes.

Finalmente se observó la imposibilidad de obtener un modelo matemático de la planta, por lo que comenzaron a realizarse multitud de experimentos para sintonizar unos parámetros PID adecuados al movimiento de cada cilindro de forma individual.

Para este apartado y los siguientes se denominará como CILINDRO 1 al cilindro izquierdo y CILINDRO 2 al derecho.

5.2. Primeras pruebas de desplazamiento.

Para llevar a cabo unos primeros experimentos del movimiento de los cilindros se prescindió de las acciones integral y derivativa de los controladores y se analizó su comportamiento al variar la constante proporcional K_p .

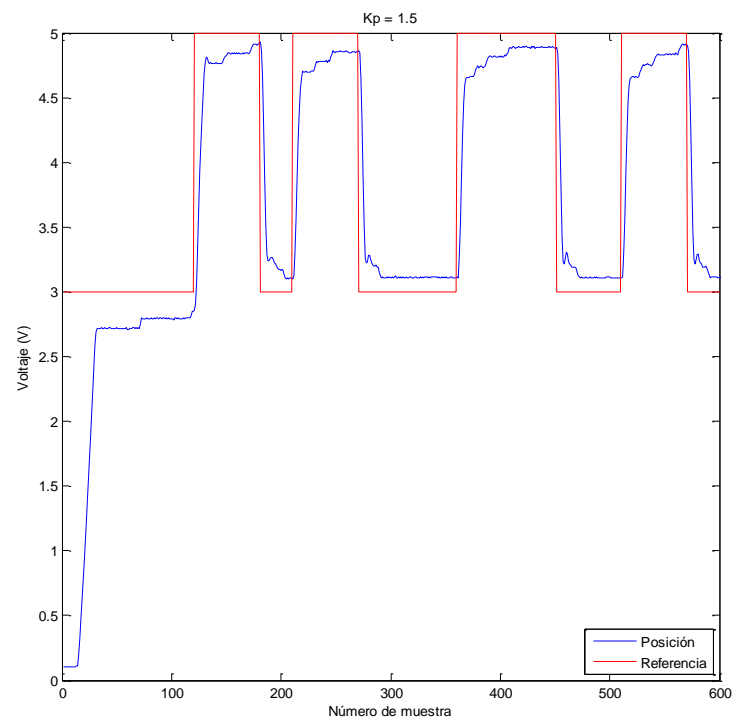
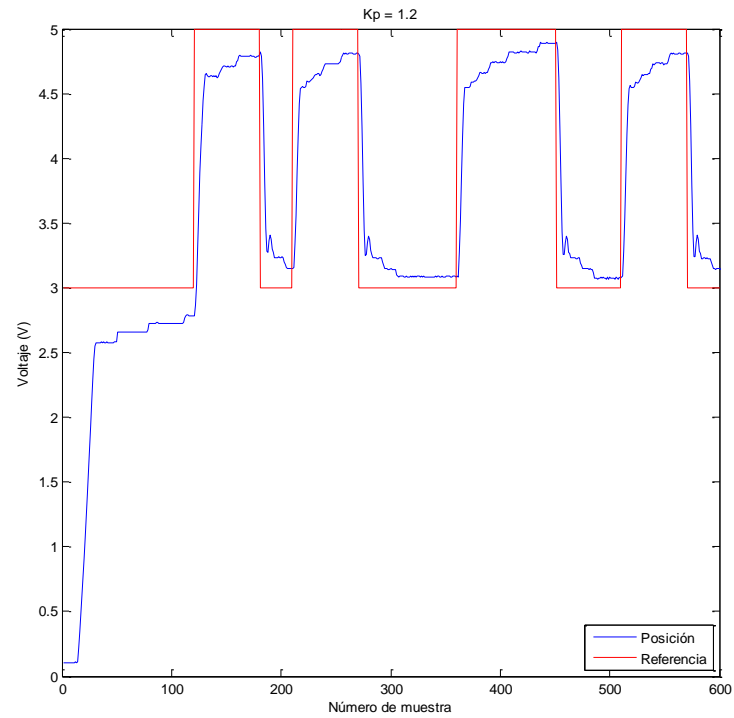


Fig. 5.1. Control de posición de un cilindro con distintos valores de la constante proporcional

Se observa que un aumento de esta constante proporcional da lugar a una respuesta con menor error en régimen permanente. Sin embargo, un incremento excesivo de este parámetro puede dar lugar a oscilaciones permanentes, es decir, a la inestabilidad del sistema.

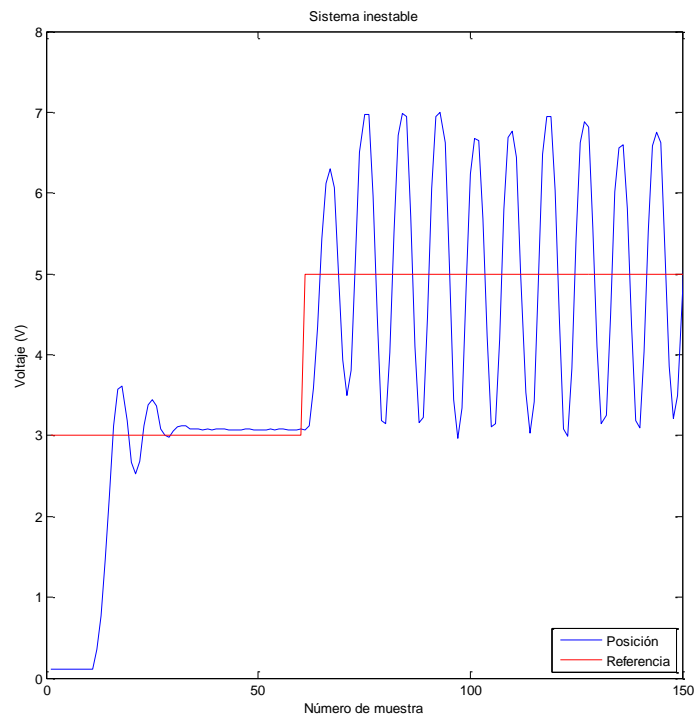


Fig. 5.2. Representación gráfica del sistema en funcionamiento inestable

Se tomó el valor de la constante proporcional como 1,5 para realizar los ensayos con los que extraer las funciones de transferencia del sistema, ya que es un valor con el que la salida se ajusta de forma aceptable a la referencia y no hay peligro de inestabilidad en caso de exigirle al sistema movimientos bruscos y variaciones importantes de posición.

5.2.1. Comparación cilindro 1 y cilindro 2.

Llevamos el cilindro 1 y el cilindro 2 a la misma referencia (en este caso, al 50% de su recorrido) y observamos la respuesta que ofrece cada uno.

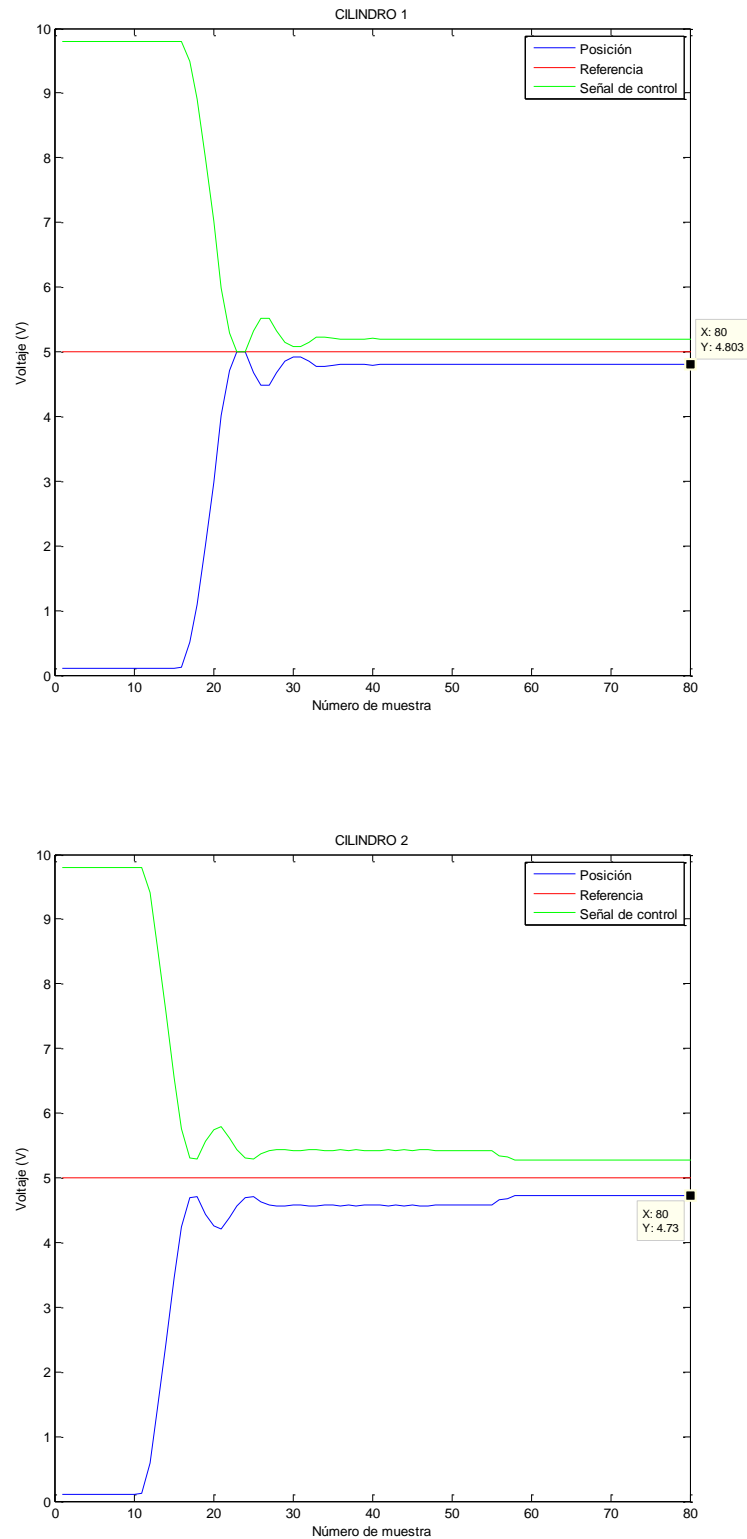


Fig. 5.3. Respuesta de los dos cilindros ante una misma situación

De esta comparación podemos extraer las siguientes conclusiones:

- El cilindro 2 tiene un mayor error estacionario.
- El cilindro 1 es más lento que el segundo, tarda más en arrancar.

5.2.2. Comparación a distintas referencias y posiciones iniciales.

La respuesta del sistema varía según la distancia que tenga que recorrer el vástago, tal y como se aprecia en las siguientes gráficas.

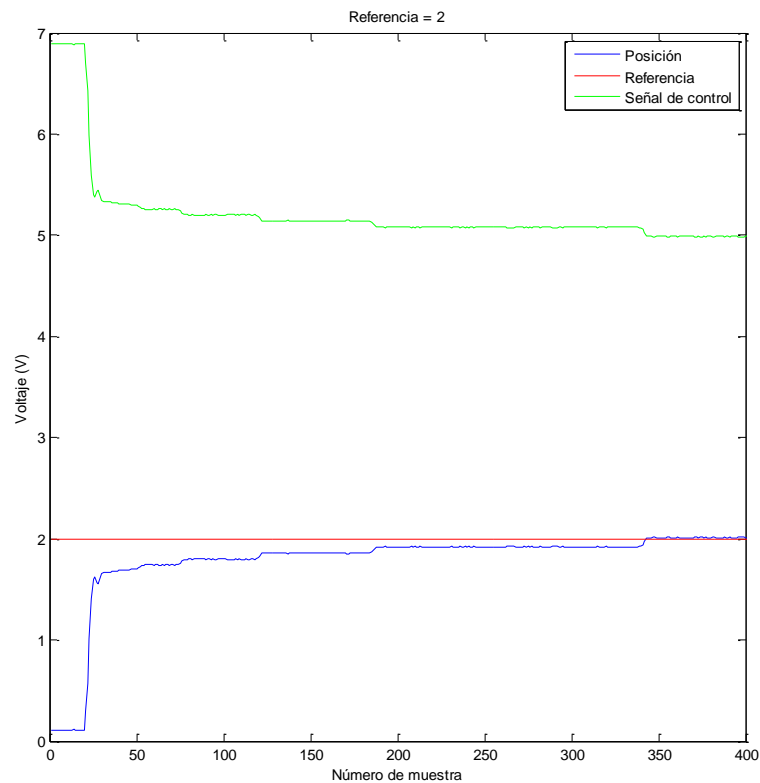


Fig. 5.4. Respuesta de un cilindro ante un desplazamiento corto

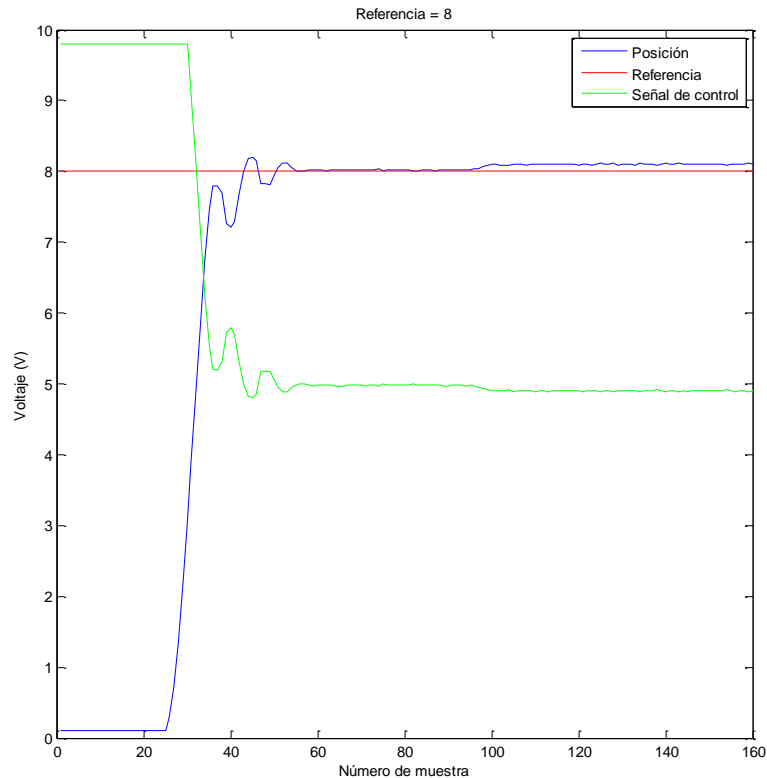


Fig. 5.5. Respuesta de un cilindro ante un desplazamiento largo

Ante un pequeño desplazamiento el sistema no presenta sobreoscilaciones, mientras que para un desplazamiento considerable del vástago éstas resultan de gran importancia y habrá que tenerlas en cuenta durante la fase de control, de forma que se reduzcan lo máximo posible y no afecten al comportamiento del otro cilindro.

Además, hay que tener en cuenta que existe una diferencia entre la respuesta del sistema ante un desplazamiento desde su posición inicial de referencia y desde una posición intermedia.

Las siguientes gráficas representan la salida que ofrecen los cilindros ante desplazamientos partiendo desde el 30% del recorrido total del vástago. En ellas se observa como la respuesta se asemeja a un sistema de segundo orden con amortiguamiento.

Tras realizar estos experimentos se llegó a la conclusión de que sería necesario tratar a cada cilindro por separado para corregir las perturbaciones inherentes a cada uno de ellos, por lo que se buscó una función de transferencia para cada cilindro que, si bien no han de ser muy distintas entre sí, deben reflejar estas pequeñas diferencias en el comportamiento de ambos cilindros.

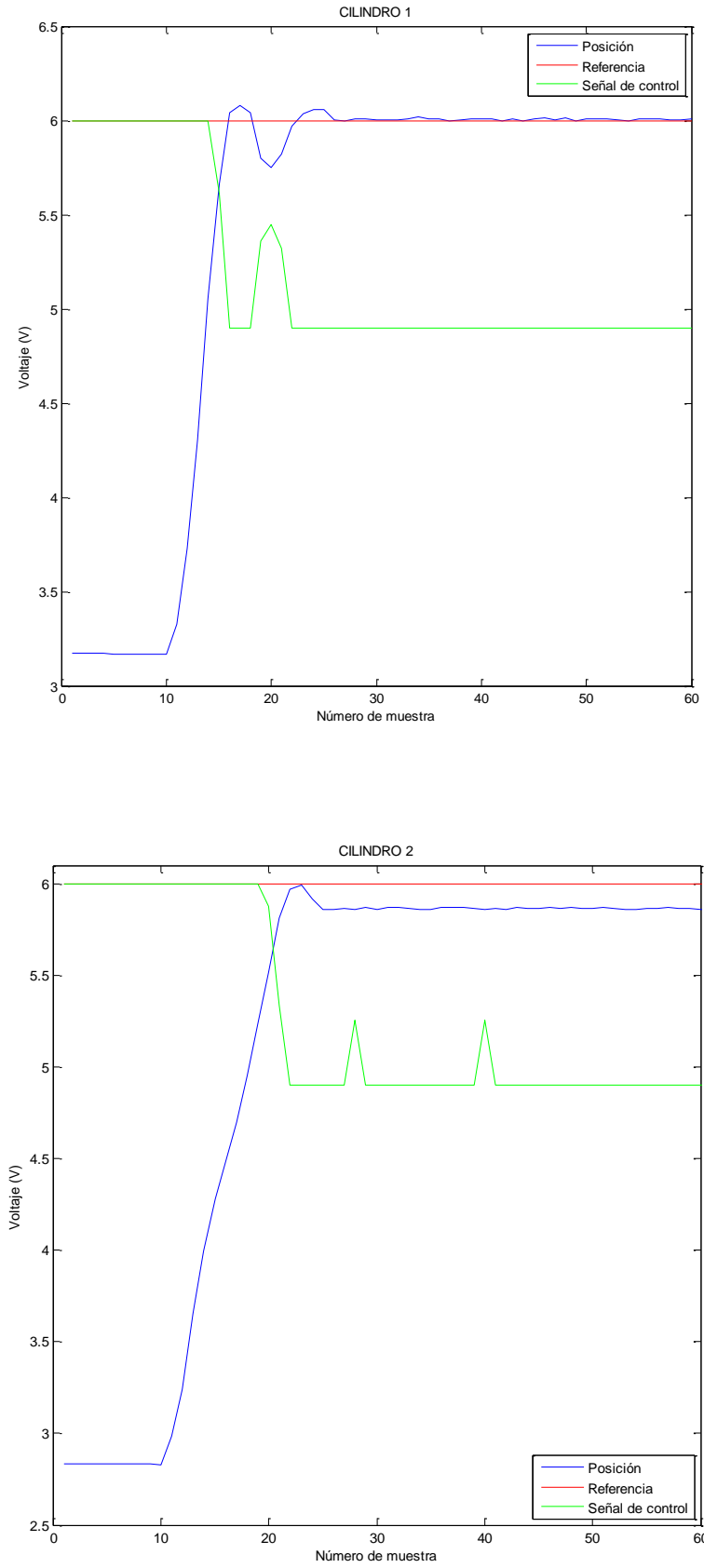


Fig. 5.6. Respuesta de ambos cilindros partiendo desde una posición intermedia hasta una referencia de 6 V

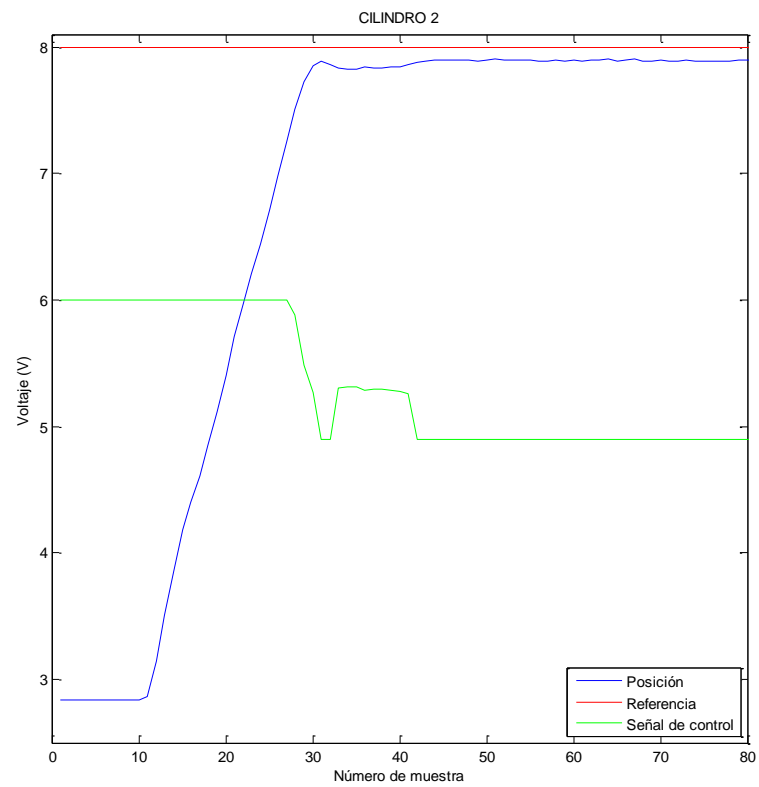
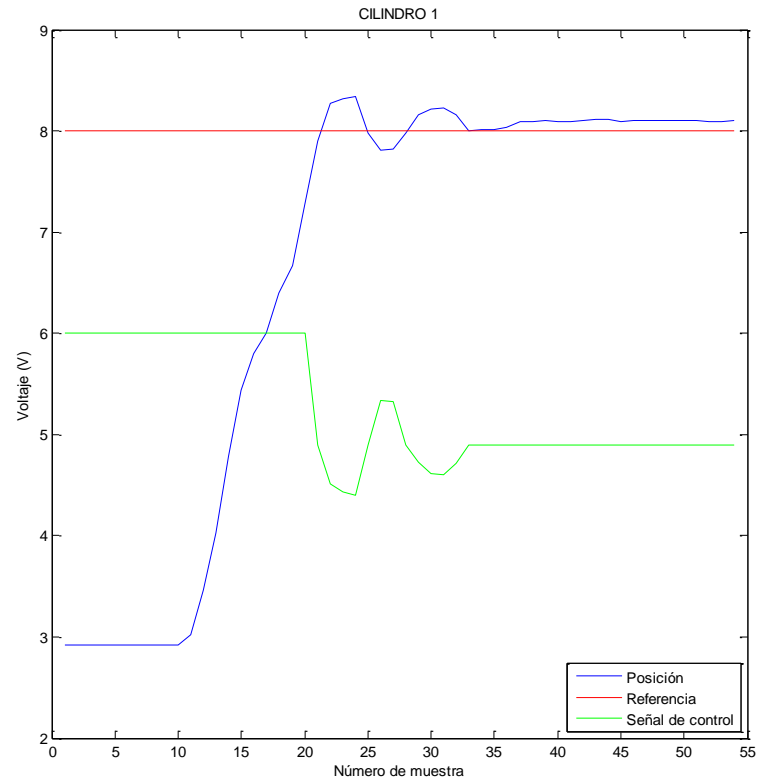


Fig. 5.7. Respuesta de ambos cilindros partiendo desde una posición intermedia hasta una referencia de 8 V

5.3. Pruebas para obtener la función de transferencia del sistema.

Para obtener las funciones de transferencia asociadas al comportamiento de ambos cilindros se analizará la respuesta que ofrecen ante diferentes entradas, mediante tres métodos.

1. Método en lazo abierto (velocidad).
2. Método en lazo cerrado (posición).
3. Método en lazo cerrado (función *ident* de *Matlab* ante entrada PRBS).

5.3.1. Método en lazo abierto.

Para realizar este experimento se pone la señal de control a un determinado valor fijo (6, 7 u 8 voltios), es decir, a diferentes aperturas de la válvula *MPYE*, lo que provoca distintas velocidades. En este método no hay ningún tipo de control, sino que el vástago se desplazará hasta el final de su carrera con mayor o menor velocidad según el valor de tensión introducido a las válvulas.

Para poder mostrar la gráfica de la velocidad respecto al tiempo, es necesario dividir los incrementos en la posición del cilindro en cada muestra entre los incrementos de tiempo, lo que en código matlab queda de la siguiente forma:

```
incdat=dat(2:length(dat))-dat(1:length(dat)-1);
%dat = vector posición en cada muestra

inct=t(2:length(t))-t(1:length(t)-1);
%t = vector tiempo en cada muestra

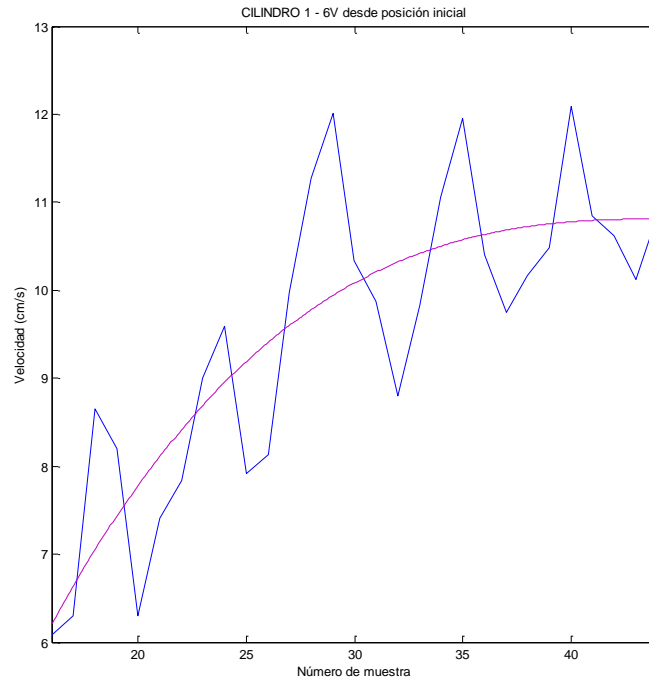
vel=incdat./inct;
```

Debido a que, tal y como se verá en los apartados sucesivos de este proyecto, se saturarán las válvulas proporcionales de forma que no reciban una tensión superior a 6 V, se representan únicamente las gráficas obtenidas a esta velocidad para ambos cilindros.

La función de transferencia se extrae aproximando las gráficas a un sistema de primer orden del que se obtiene la ganancia y la muestra para la que el sistema se empieza a mover.

Si denominamos K a la ganancia (valor estacionario) y n_0 a la muestra inicial, la función de transferencia de un sistema de primer orden será de la siguiente forma:

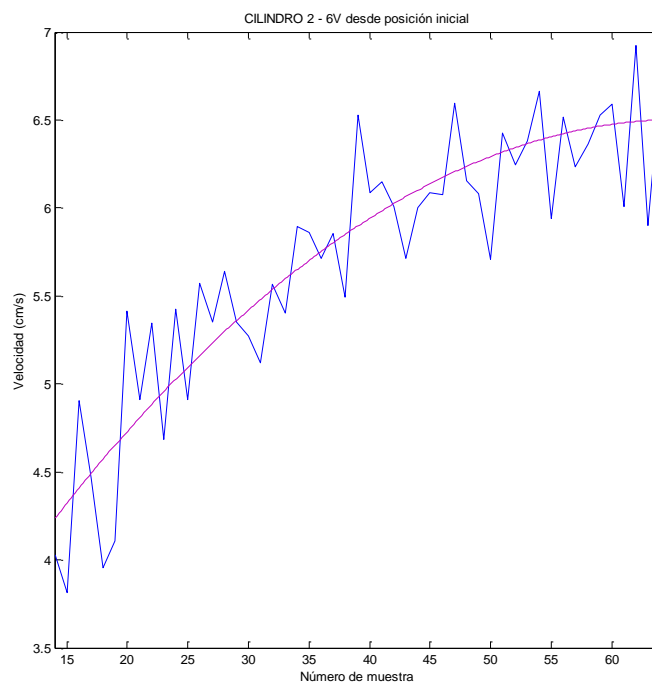
$$t.f. = \frac{K}{1 + \left(\frac{n_0}{30}\right)s}$$



$$K = 10.84$$

$$n_0 = 15$$

Fig. 5.8. Cilindro 1. Método en lazo abierto. Sistema de primer orden



$$K = 6.5$$

$$n_0 = 13,4$$

Fig. 5.9. Cilindro 2. Método en lazo abierto. Sistema de primer orden

Como se ha demostrado anteriormente, el sistema no se comporta de igual forma si el vástago sale desde su posición de referencia o si éste se encuentra en una posición intermedia, ya que en este caso la resistencia a vencer es menor. Por ello se tomará el 30% del recorrido del vástago como posición inicial y se comprobará que las gráficas de velocidad se asemejan a un sistema de segundo orden.

En un sistema de segundo orden, la función de transferencia viene dada por:

$$t.f. = \frac{K w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

donde K es la ganancia o valor de la función en régimen estacionario, y los parámetros ζ (coeficiente de amortiguamiento) y w_n (frecuencia natural) se obtienen de la siguiente forma:

$$\text{El máximo sobrepico se calcula como } Mp = \frac{\text{Valor sobrepico} - \text{Valor estacionario}}{\text{Valor estacionario} - \text{Valor inicial}}$$

Partiendo de la fórmula que relaciona el máximo pico con el coeficiente de amortiguamiento,

$$Mp = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\text{despejamos éste último: } \zeta = \frac{(\ln(Mp))^2}{\pi^2 + (\ln(Mp))^2}$$

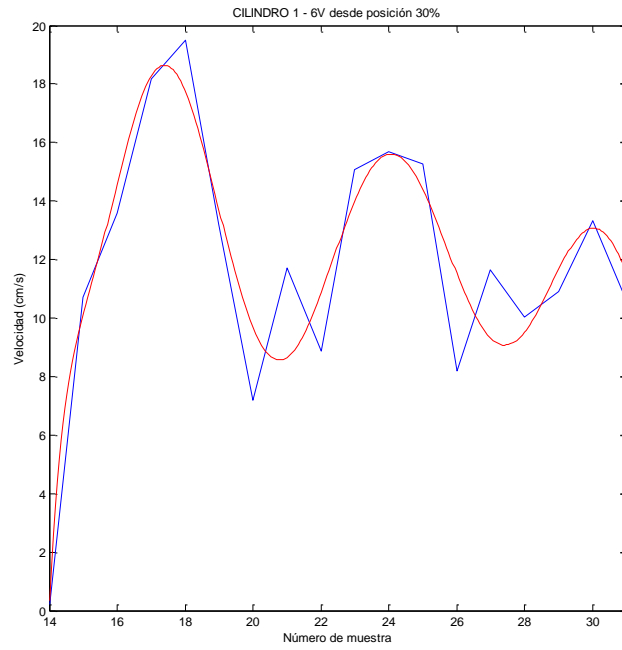
La frecuencia natural se obtiene a partir del tiempo de pico, t_p .

$$t_p = \frac{\text{Número de muestras hasta el sobrepico}}{\text{Número de muestras por segundo}}$$

En este proyecto, el denominador será igual a 30 muestras por segundo.

De ahí obtenemos la frecuencia forzada, $w_d = \frac{\pi}{t_p}$

Y la frecuencia natural es $w_n = \frac{w_d}{\sqrt{1-\zeta^2}}$



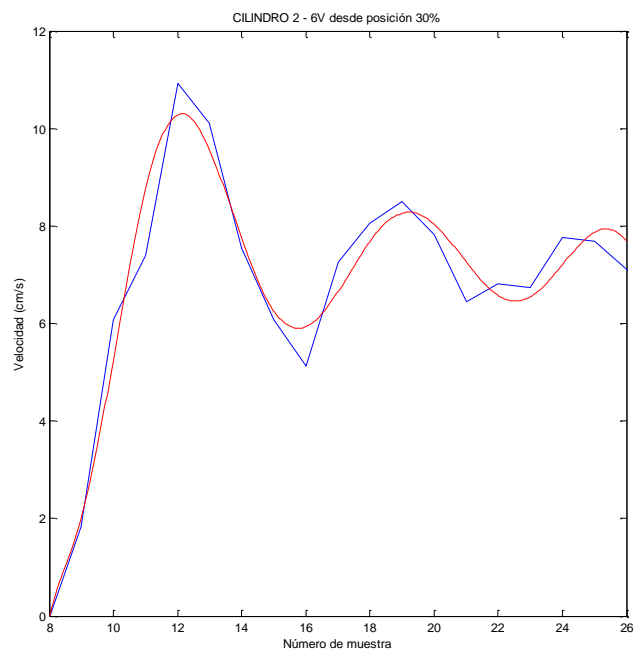
$$K = 11.94$$

$$M_p = 0.63$$

$$\zeta = 0.14$$

$$\omega_n = 23.81$$

Fig. 5.10. Cilindro 1. Método en lazo abierto. Sistema de segundo orden



$$K = 7$$

$$M_p = 0.56$$

$$\zeta = 0.18$$

$$\omega_n = 23.96$$

Fig. 5.11. Cilindro 2. Método en lazo abierto. Sistema de segundo orden

Finalmente, las funciones de transferencia obtenidas por este método se recogen en la siguiente tabla:

LAZO ABIERTO	CILINDRO 1	CILINDRO 2
Desde posición inicial = 0	$\frac{10.8}{1 + 0.5s}$	$\frac{6.5}{1 + 0.4s}$
Desde 30% del recorrido	$\frac{6768.9}{s^2 + 6.9s + 566.9}$	$\frac{4018.5}{s^2 + 8.7s + 574.1}$

Estos experimentos en lazo abierto no se pueden considerar muy fiables, como veremos más adelante, debido a posibles errores en los incrementos de tiempo, a ruidos o a algún dato atípico surgido durante las pruebas.

5.3.2. Método en lazo cerrado.

Durante la ejecución de estas pruebas existe un control PID de la posición del cilindro en todo momento, de modo que podemos indicar el punto de la carrera al que queremos llevar el vástago.

Las pruebas en lazo cerrado se realizaron para una posición inicial del vástago igual al 30% de su recorrido (9 cm), ya que los cilindros partirán desde una posición intermedia prácticamente en todos los movimientos que se lleven a cabo durante el seguimiento de trayectorias. Por este motivo en este apartado se ha centrado la atención sobre este aspecto, pues si se controla adecuadamente, el control del cilindro partiendo desde su posición inicial no dará lugar a mayores problemas.

5.3.2.1. Método gráfico.

Consiste en extraer las gráficas de la posición con respecto al tiempo o al número de muestra (ambos parámetros se relacionan mediante la frecuencia de muestreo) para una referencia cualquiera. La función de transferencia se obtiene aproximando la salida a un sistema de segundo orden tal y como se explica en el apartado anterior de lazo abierto.

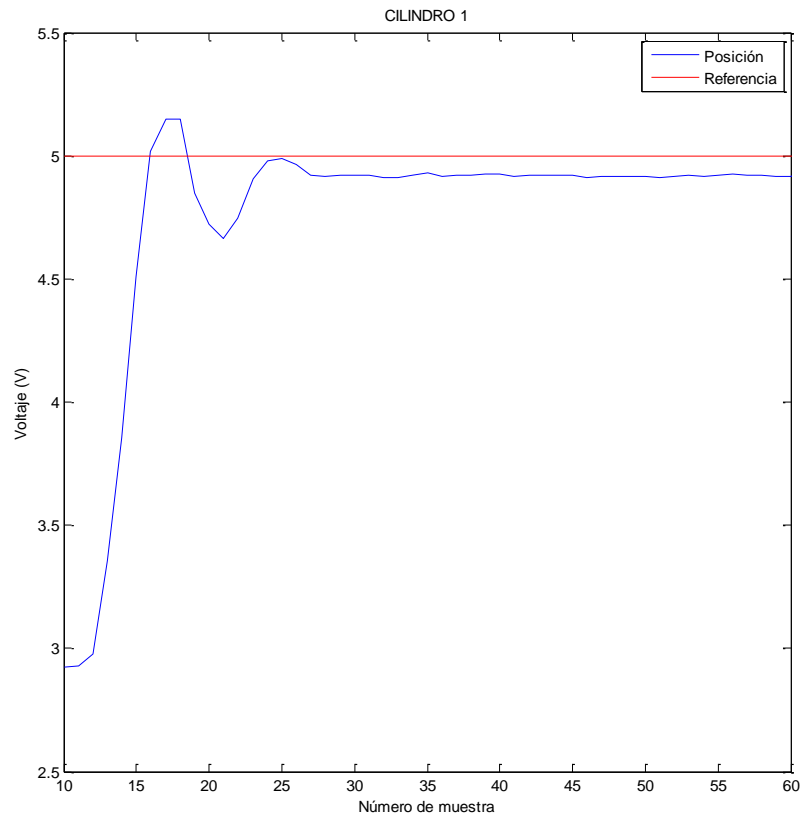
En este caso la función de transferencia es la del lazo, que para un controlador de tipo proporcional, como es el caso de estas pruebas realizadas, tiene la siguiente forma:

$$t.f. = \frac{K_p G(s)}{1 + K_p G(s)}$$

De esta expresión hay que extraer la función $G(s)$ asociada al sistema en lazo abierto.

Las pruebas han sido realizadas, tal y como se ha comentado en este mismo capítulo y en anteriores, a una presión de 5 bar y con una constante proporcional del controlador $K_p = 1.5$.

A continuación se representan las gráficas más significativas de ambos cilindros desplazados desde el 30% del recorrido total del vástago hasta una referencia igual al 50% del mismo.



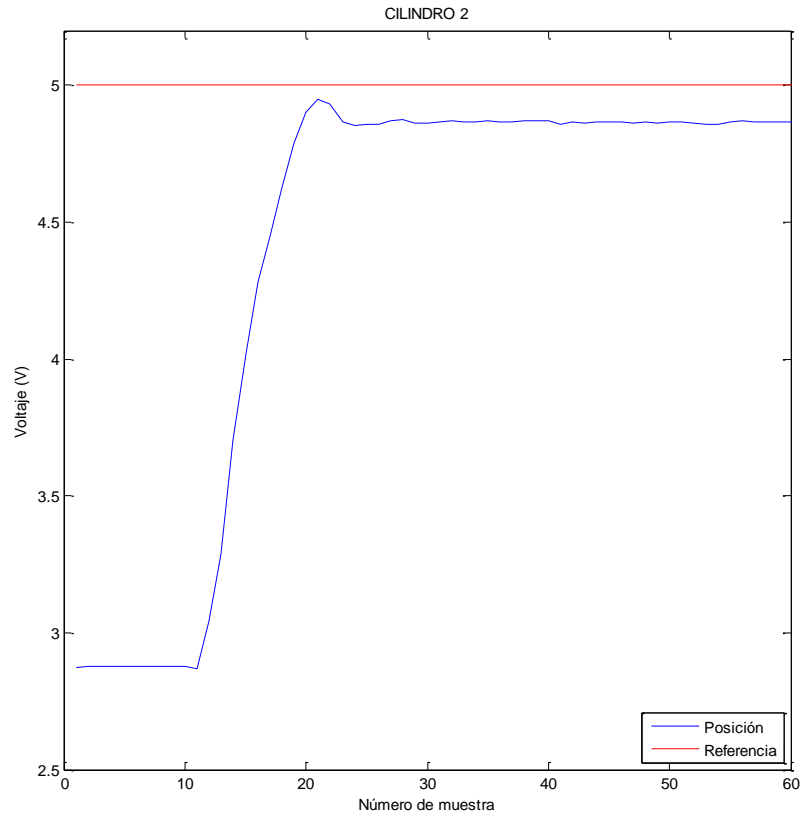
$$M_p = 0.14$$

$$\zeta = 0.53$$

$$\omega_n = 15.88$$

$$t.f. = \frac{252.2}{s^2 + 16.8s + 252.2}$$

Fig. 5.12. Cilindro 1. Método gráfico en lazo cerrado



$$M_p = 0.02$$

$$\zeta = 0.77$$

$$\omega_n = 11.40$$

$$t.f. = \frac{259.8}{s^2 + 17.6s + 259.8}$$

Fig. 5.13. Cilindro 2. Método gráfico en lazo cerrado

Las funciones de transferencia más representativas de cada cilindro y obtenidas de esta forma se recogen en la siguiente tabla:

LAZO CERRADO	CILINDRO 1	CILINDRO 2
Método gráfico	$\frac{9.98}{1 + 0.059s}$	$\frac{9.85}{1 + 0.057s}$

5.3.2.2. Mediante entrada vector de pulsos PRBS.

El último método para extraer una función de transferencia aproximada del sistema consiste en la introducción de un vector de impulsos de periodo aleatorio PRBS. Se toman los datos de la respuesta de la posición de los cilindros y mediante la función *ident* de Matlab se obtiene una función de transferencia del sistema.

Este experimento se ha realizado en lazo cerrado entre los valores de referencia 3 y 5, es decir, entre el 30% y el 50% de la carrera del vástago ya que de no ser así, en cada impulso se alcanzaría el final del recorrido y el comportamiento de los cilindros no sería lineal.

A pesar de evaluar el sistema en lazo cerrado, la función *ident* proporciona una función de transferencia en lazo abierto, por lo que no hay que realizar la conversión explicada en el apartado anterior.

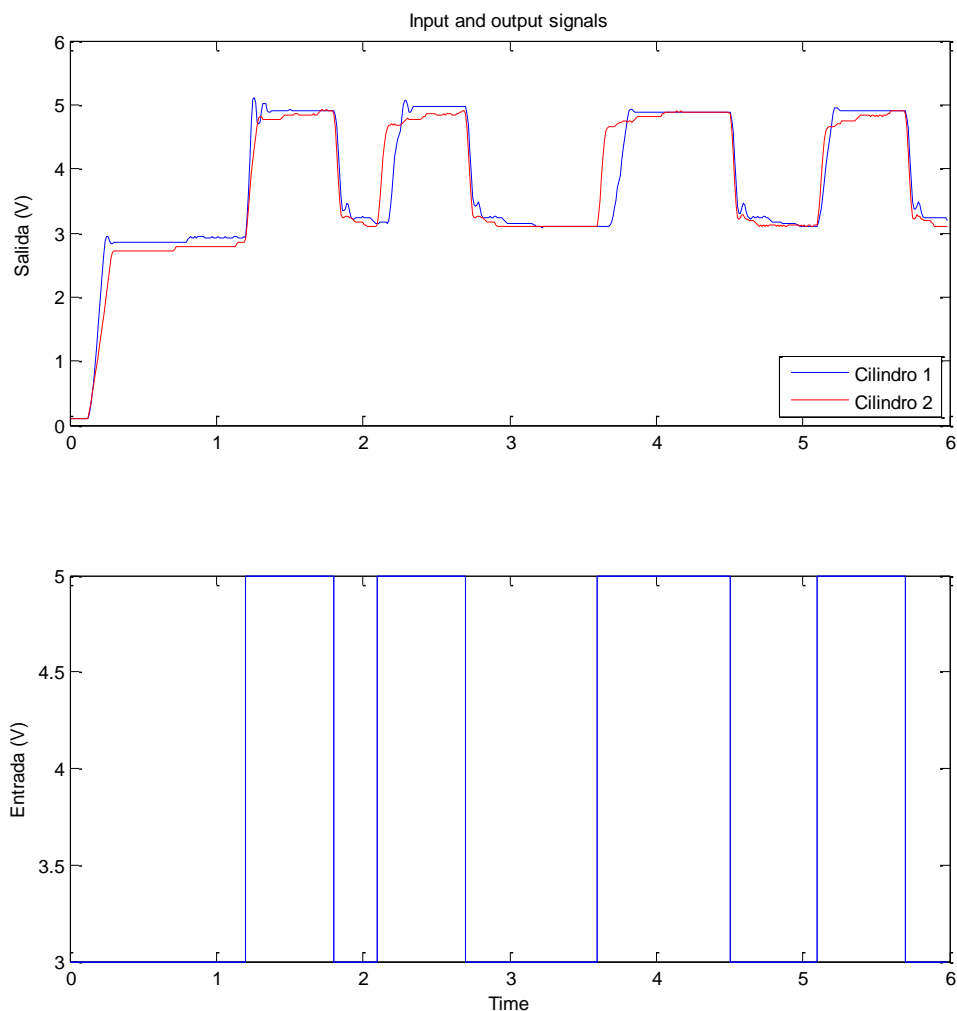


Fig. 5.14. Señales de entrada y salida de ambos cilindros

Las funciones de transferencia que mejor se ajustan al comportamiento del sistema obtenidas por este método se recogen en la siguiente tabla:

LAZO CERRADO	CILINDRO 1	CILINDRO 2
Método <i>ident</i> (Matlab)	$\frac{0.990}{1 + 0.083s}$	$\frac{0.976}{1 + 0.056s}$

Y la respuesta que ofrecen ante una entrada escalón unitario es la siguiente:

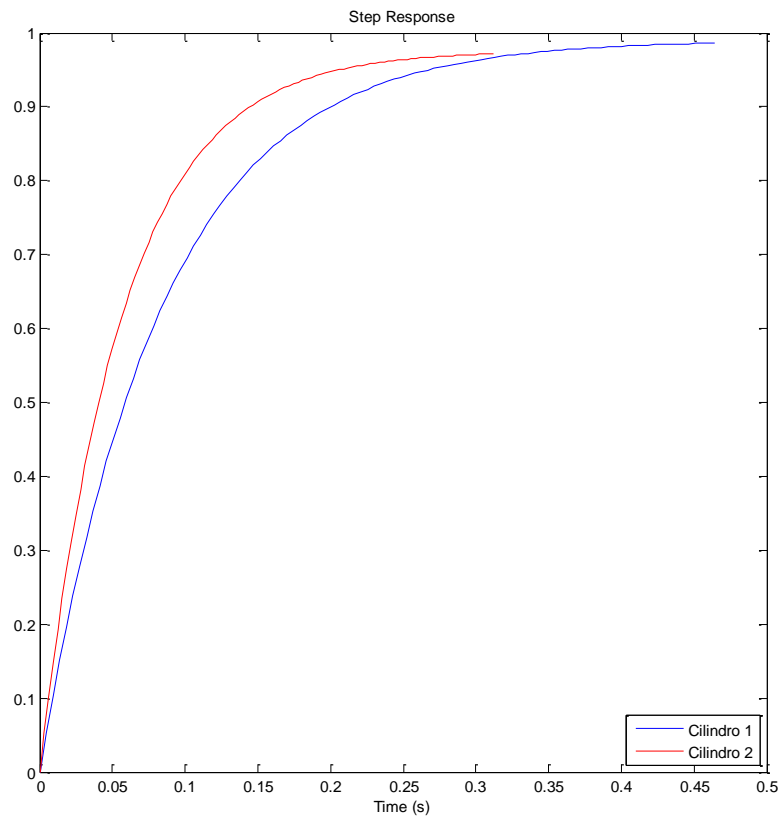


Fig. 5.15. Respuesta de ambos cilindros ante una entrada de tipo escalón unitario

5.4. Simulaciones gráficas de las funciones de transferencia obtenidas.

Una vez obtenidas las distintas funciones de transferencia asociadas a cada uno de los cilindros es necesario comprobarlas para ver qué métodos se ajustan mejor al comportamiento del sistema y, además, cuantificar las diferencias existentes entre la respuesta de cada uno de los cilindros ante la misma entrada.

5.4.1. Comparaciones entre métodos.

Tal y como se ha comentado anteriormente, los métodos empleados para la obtención de funciones de transferencia han sido muy diversos y no todos ellos reflejan correctamente el comportamiento del sistema.

A continuación se representan las respuestas de las funciones de transferencias del cilindro 1 ante una entrada de tipo escalón unitario. En la gráfica se observa una clara diferencia entre los métodos en lazo abierto correspondientes a una posición inicial del vástago igual a su posición de referencia y una posición inicial igual al 30% del recorrido del mismo.

Sin embargo, los métodos en lazo cerrado ofrecen respuestas muy similares que se ajustan adecuadamente al comportamiento real del cilindro. Los métodos en lazo abierto de velocidad, como ya se ha comentado, no son muy fiables, por los que de ahora en adelante no se tendrán en cuenta para el análisis del sistema.

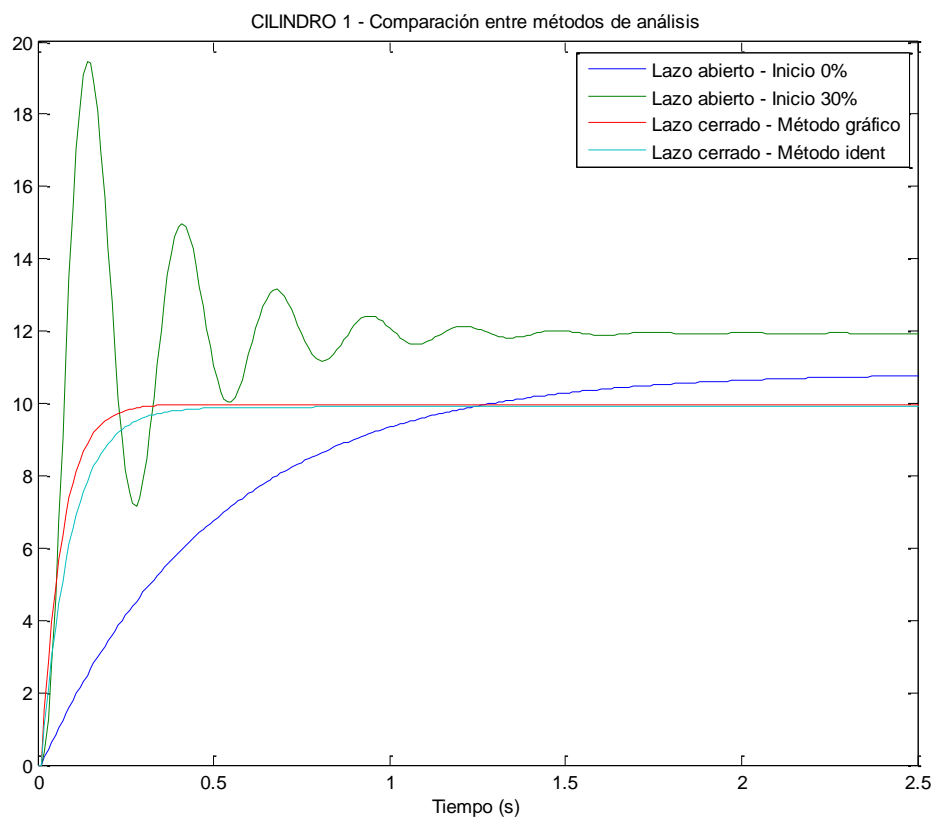


Fig. 5.16. Representación gráfica de la respuesta obtenida por cada método de análisis

Si se realiza la misma comparación para el cilindro 2 se obtienen resultados análogos, por lo que se llega a la conclusión de que los métodos en lazo cerrado (tanto el método gráfico como el que involucra al vector de pulsos PRBS) son los más apropiados para realizar el análisis de las funciones de transferencia del sistema.

5.4.2. Comparaciones entre cilindros.

Las pruebas realizadas de desplazamiento de los cilindros nos ofrecían una primera impresión de la diferencia en el comportamiento de ambos cilindros, lo cual se ve reflejado claramente tras realizar las simulaciones pertinentes en *Simulink*.

Se representa en una misma gráfica la respuesta de ambos cilindros ante una entrada escalón unitario y se observan sus características.

Cabe destacar que las conclusiones que se obtienen son las mismas tanto si se representan las funciones de transferencia obtenidas mediante el método gráfico en lazo cerrado como si se hace lo propio con las obtenidas mediante la función *ident* de Matlab.

En este caso se han evaluado las funciones de transferencia extraídas mediante la función *ident* y el vector de pulsos.

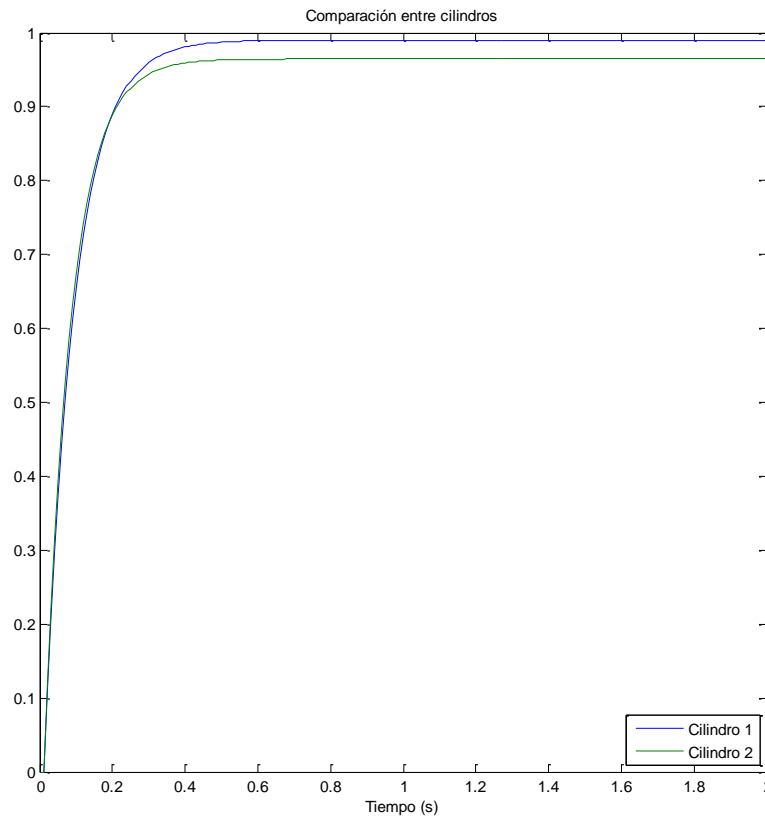


Fig. 5.17. Respuesta de cada cilindro ante una entrada de tipo escalón unitario

Observando la figura se aprecia que las conclusiones extraídas a priori acerca del comportamiento de los cilindros son, efectivamente, correctas. Esto es, el cilindro izquierdo tiene mayor ganancia que el segundo y tarda más en llegar a la referencia indicada. Estos aspectos son de suma importancia a la hora de implementar el controlador PID que mejor comportamiento otorgue al sistema.

5.5. Sintonización de los controladores PID.

Para determinar los valores de la ganancia proporcional K_p , del tiempo integral T_i y del tiempo derivativo T_d en base a las características de la respuesta transitoria de una planta existen varios métodos. Los más utilizados son las reglas de sintonización de Ziegler-Nichols, dos métodos en los que se pretende obtener un 25% máximo de sobrepico ante una entrada escalón.

En el primer método, la respuesta de la planta a una entrada escalón se obtiene de manera experimental. Si la planta no contiene integradores ni polos dominantes complejos conjugados, la curva de la respuesta puede tener forma de S , como se observa en la figura.

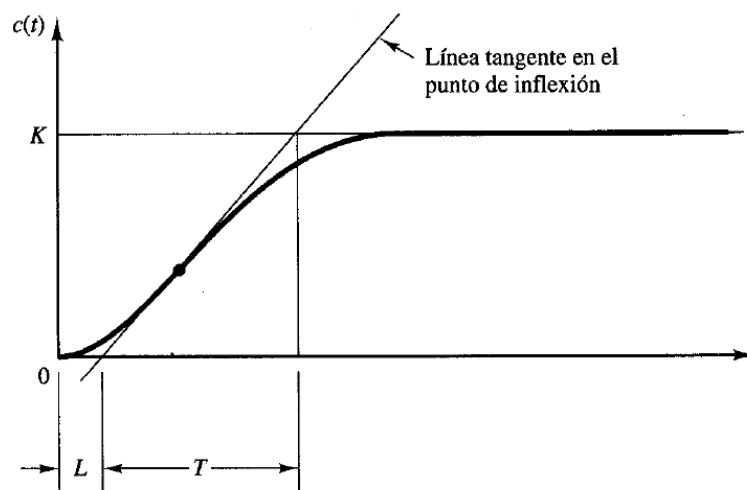


Fig. 5.18. Curva para aplicar el primer método de Ziegler-Nichols

En las gráficas anteriores se observa claramente que la gráfica no presenta forma de S lo que concuerda con el hecho de que el sistema presenta un integrador. Por tanto este método no es aplicable a la planta en estudio.

En el segundo método se establece $K_i=K_d=0$ y, usando sólo la acción de control proporcional, se incrementa K_p hasta un valor crítico K_{cr} para el cual la salida exhiba oscilaciones sostenidas, del siguiente modo:

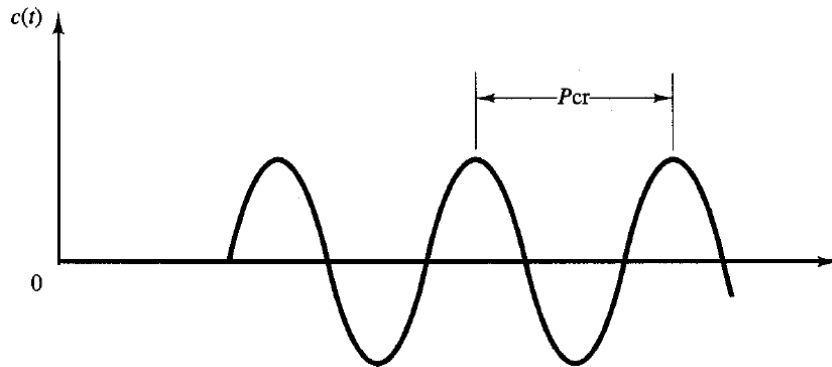


Fig. 5.19. Gráfica de aplicación del segundo método de Ziegler-Nichols

Tras realizar múltiples pruebas en lazo cerrado no se ha podido hallar la ganancia crítica que provoca la situación anterior, de forma que este segundo método de Ziegler-Nichols tampoco es aplicable a este sistema.

Estas reglas de sintonización se han usado ampliamente para sintonizar controladores PID en los sistemas de control de procesos en los que se conoce la dinámica de la planta. Cuando esto no es posible, se recurre a métodos experimentales, como ensayos de prueba y error, para obtener los parámetros del controlador PID.

5.5.1. Sintonización experimental de controladores.

Debido a la imposibilidad de implementar los controladores por medio de los métodos mencionados anteriormente es necesario recurrir a pruebas experimentales. Para ello se llevaron a cabo numerosos desplazamientos de los cilindros modificando levemente los parámetros del controlador para obtener la respuesta que mejor se ajuste a las referencias indicadas.

A continuación se muestran algunas de las pruebas realizadas con diferentes parámetros del controlador PID.

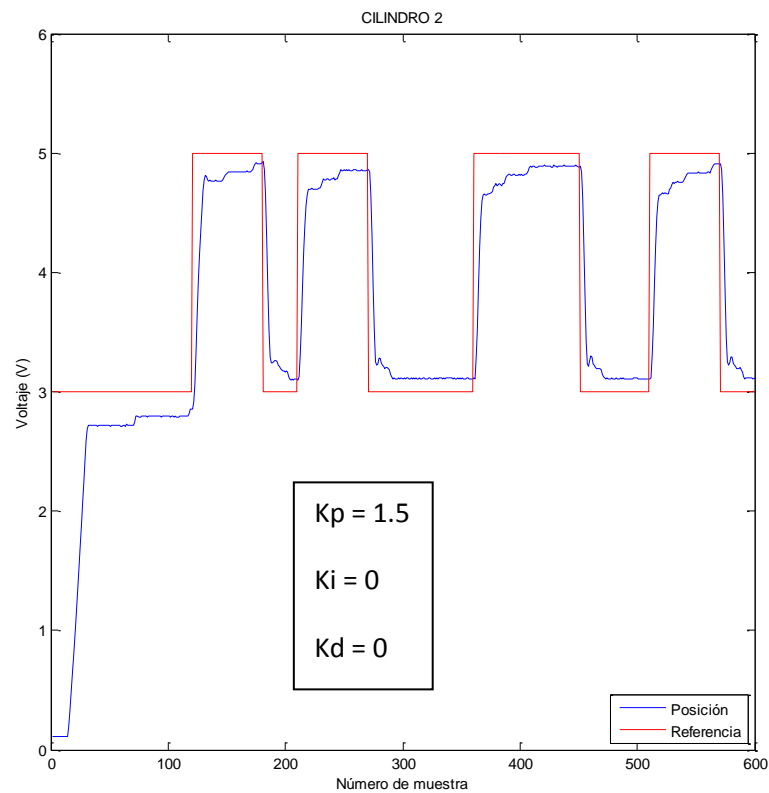
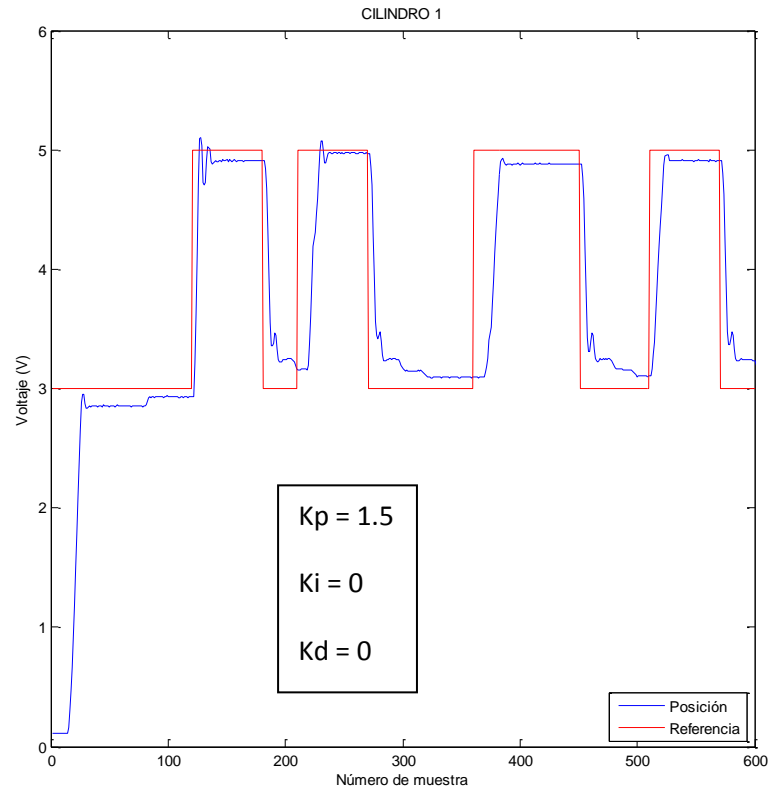


Fig. 5.20. Respuesta de los cilindros sin acción proporcional e integral

Se aprecia un error de posición considerable, por lo que se aumenta la constante proporcional y se incluye una acción integral para tratar de limitarlo.

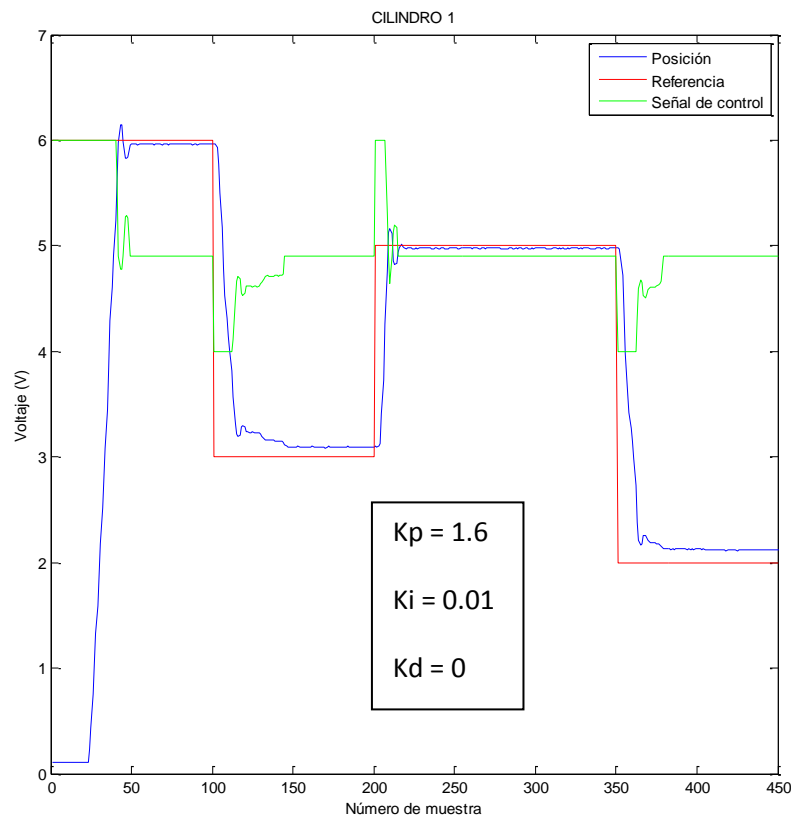


Fig. 5.21. Respuesta de un cilindro con acción proporcional e integral

Ahora que el error de posición se ha reducido bastante conviene incluir una acción derivativa en el sistema para minimizar las vibraciones que se dan en el cilindro al desplazarse desde una referencia a otra superior. Asimismo, se incrementarán las acciones proporcional e integral para evitar que el sistema se haga inestable.

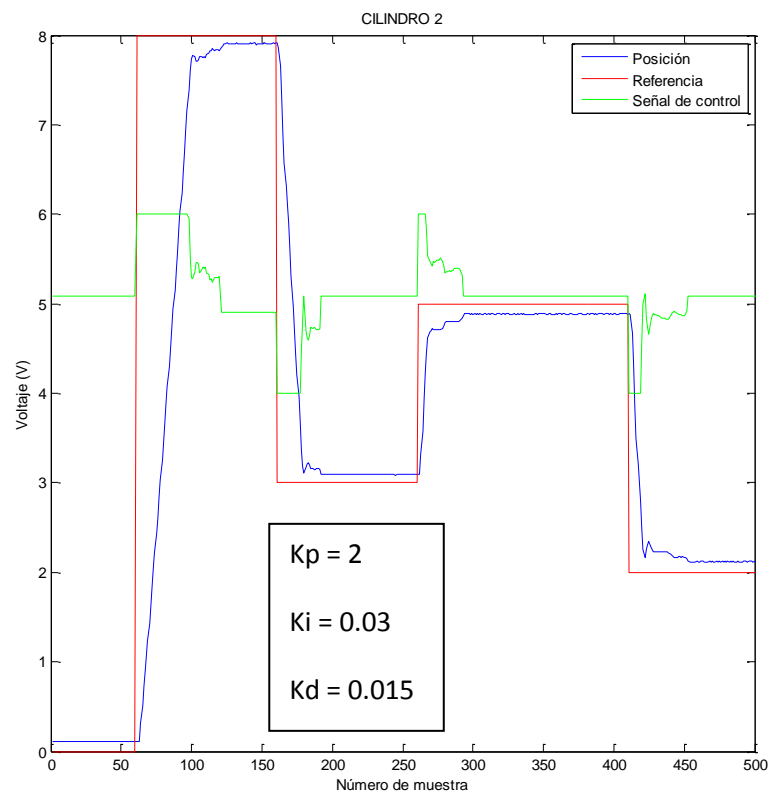
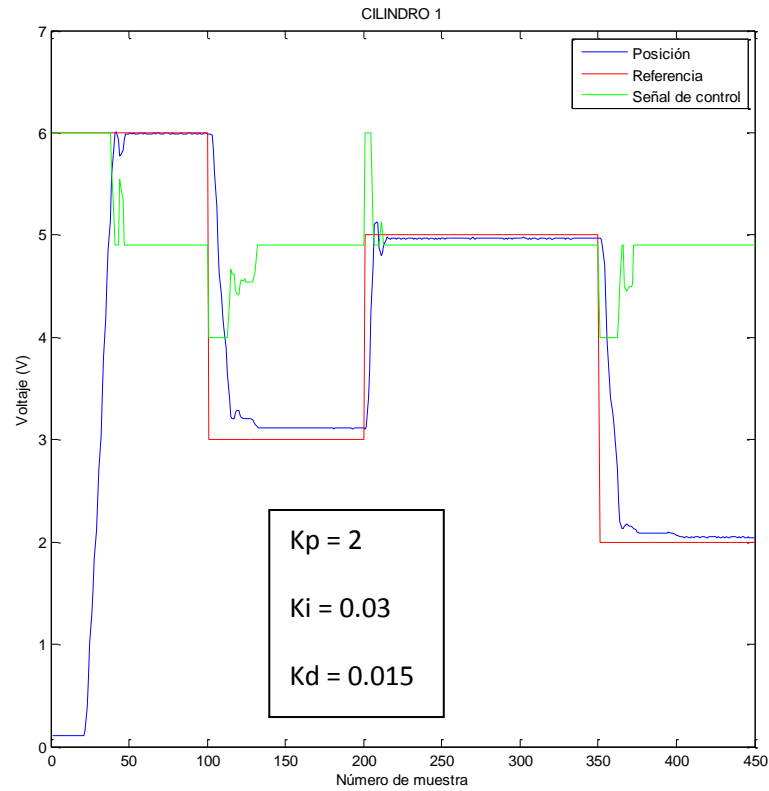


Fig. 5.22. Respuesta de ambos cilindros con los controladores PID sintonizados

Se recogen en la siguiente tabla los parámetros de los controladores PID que se han obtenido.

PARÁMETROS PID	CILINDRO 1	CILINDRO 2
Kp	2.0	2.0
Ki	0.03	0.03
Kd	0.015	0.015

La respuesta que obtenemos con estos parámetros es lo suficientemente precisa para poder analizar el movimiento conjunto de ambos cilindros y las perturbaciones que ejerce el uno sobre el otro. Cabe destacar que, si bien ambos controladores son iguales en este momento, cuando se analice el movimiento simultáneo de los cilindros será necesario tener en cuenta las diferencias de comportamiento existentes, en forma de ganancia proporcional y oscilaciones no deseadas.

CAPÍTULO 6:
CONTROL SIMULTÁNEO DE LOS
CILINDROS.

6.1. Introducción.

Tras las pruebas realizadas de los controladores PID individuales se ha continuado el desarrollo desplazando los dos cilindros a la vez. Si se le aplica el mismo nivel de tensión a ambos se observa que el izquierdo avanza antes que el derecho, de forma que para obtener un movimiento simultáneo será necesario aplicar más tensión a éste último o aumentar su ganancia.

Los controladores de cada cilindro no provocan la misma respuesta en ellos si se mueven a la vez ya que cada cilindro al desplazarse provoca una perturbación en el otro. El sistema pasa ahora a ser *MIMO* (multiple input-multiple output) y su estudio resulta más complejo, hasta el punto de que en el presente proyecto se han empleado exclusivamente métodos experimentales. El esquema de este tipo de sistemas es el siguiente:

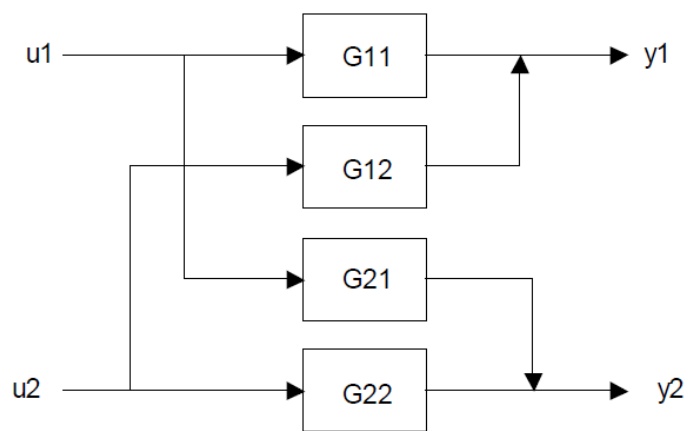


Fig. 6.1. Esquema de perturbaciones de un sistema MIMO

Para poder realizar el desplazamiento simultáneo los programas informáticos del control de cada cilindro se implementan en uno solo y se distinguen las variables correspondientes a los parámetros de cada controlador mediante un vector de dos índices, uno para cada cilindro.

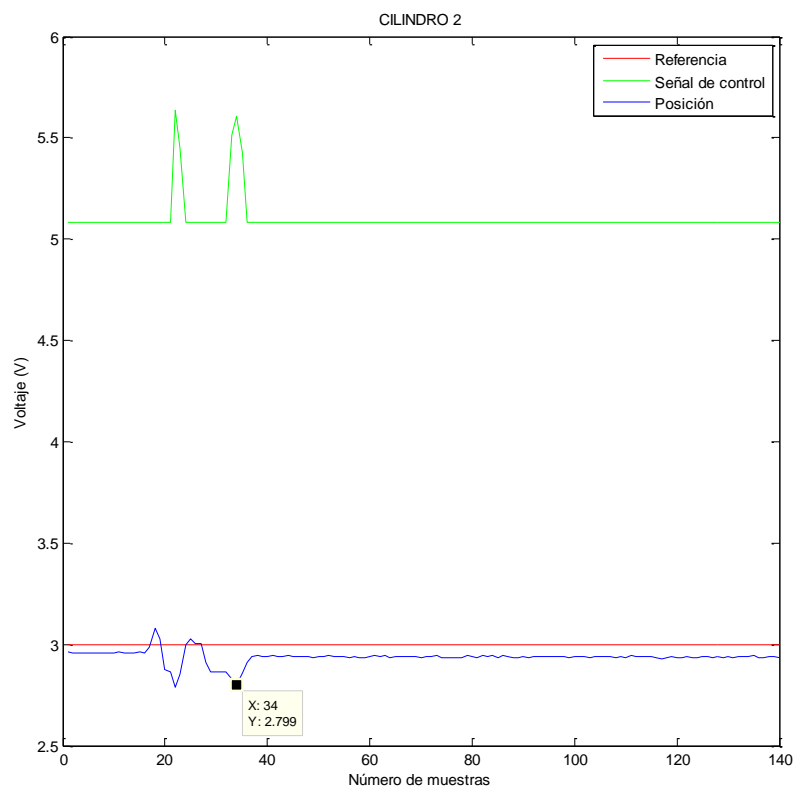
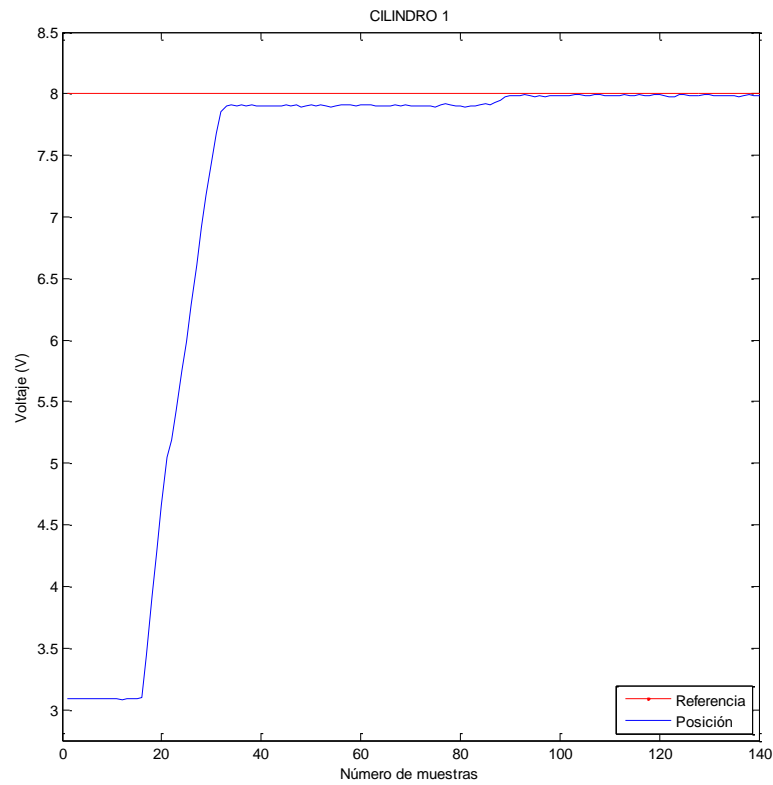
En este capítulo se describen las pruebas realizadas para sintonizar los nuevos parámetros de los controladores y sincronizar los cilindros, esto es, conseguir que ambos se desplacen a la misma velocidad y al mismo tiempo.

6.2. Perturbaciones asociadas al desplazamiento de los cilindros.

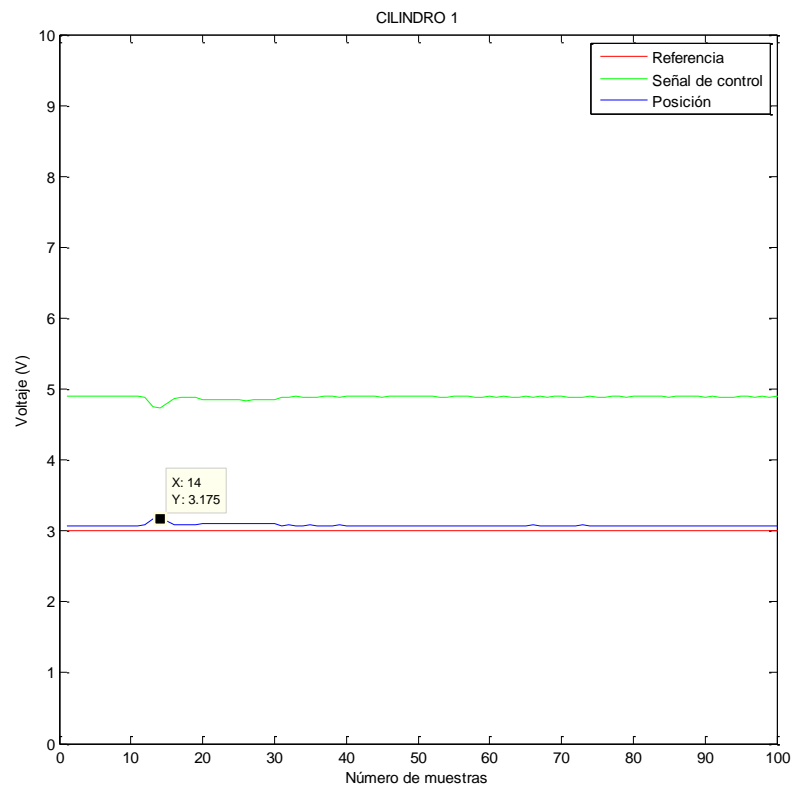
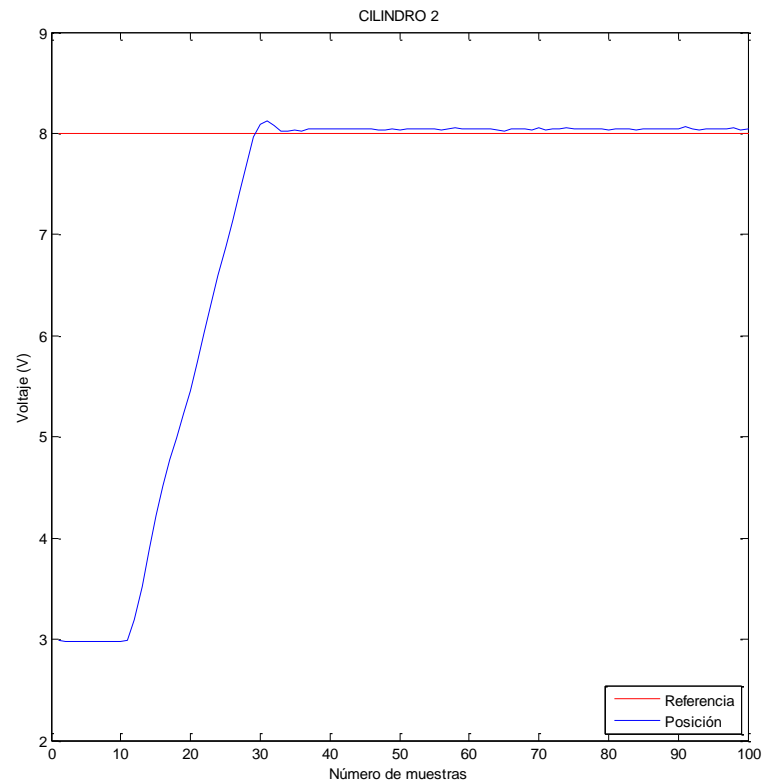
Tal y como se ha comentado, el movimiento de un cilindro afecta al otro. Con el fin de cuantificar estas perturbaciones se han realizados experimentos consistentes en dejar uno de los cilindros inmóvil y aplicar al distintos valores de referencia al otro.

De este modo, dejando el cilindro derecho en una posición fija igual al 30% de su recorrido y desplazando el izquierdo, se obtienen las siguientes figuras.

- Perturbación del cilindro 2 debido al desplazamiento del cilindro 1.



- Perturbación del cilindro 1 debido al desplazamiento del cilindro 2.

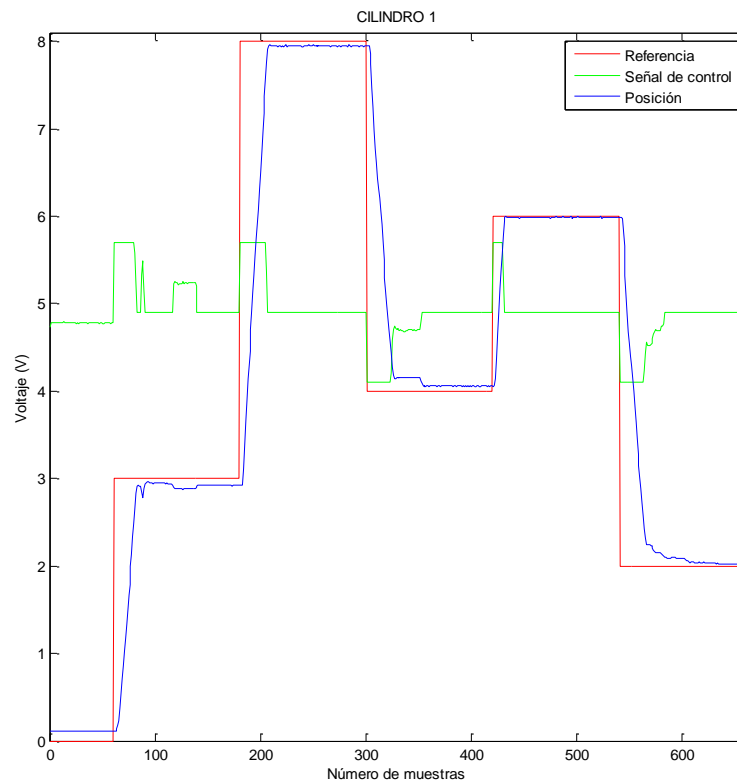


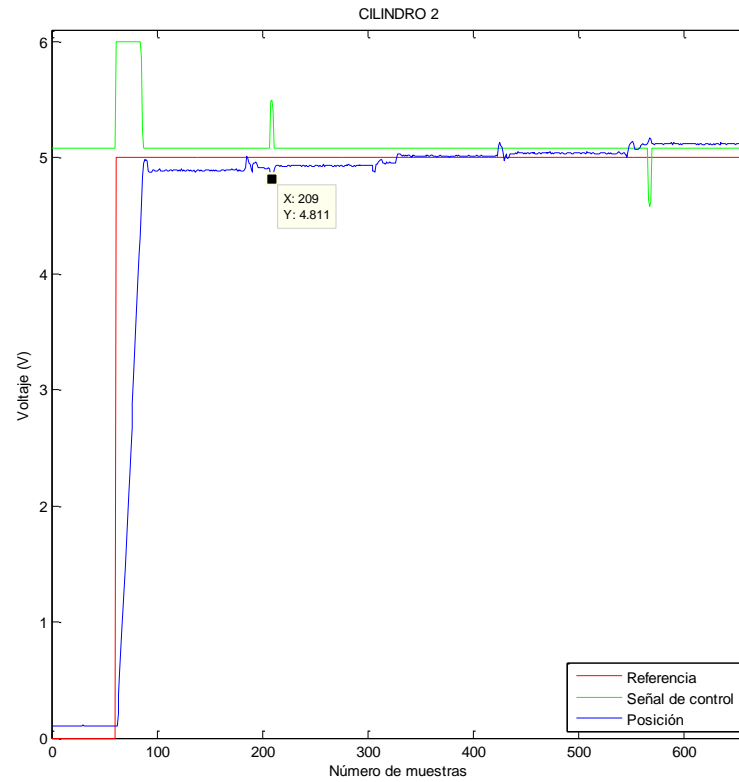
Como se puede apreciar, el cilindro número 2, el derecho, se ve más afectado por los movimientos externos que el primero. Esto puede ser debido a muchas causas, entre ellas la menor rigidez estructural que presenta tanto en su base como en su punto superior.

Para tratar de corregir estas perturbaciones se llevó a cabo el proceso de sincronización y saturación del caudal de aire en las válvulas, que será explicado en este mismo capítulo más adelante, además de las pruebas para sintonizar los parámetros adecuados de los controladores PID.

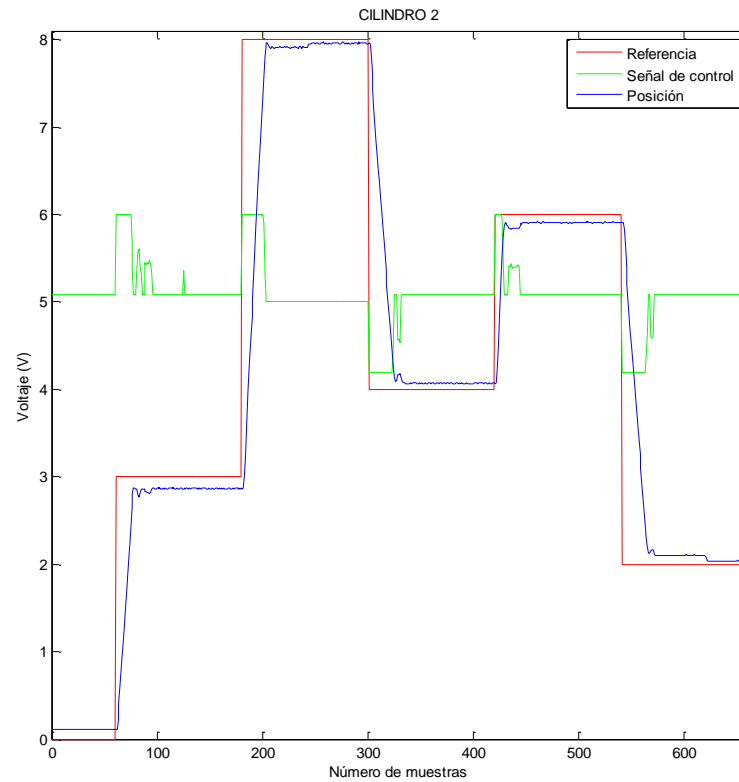
Una vez realizado se observa una reducción drástica de las perturbaciones que cada cilindro origina en el opuesto, aunque nunca llegan a eliminarse por completo, como se puede ver en las siguientes figuras.

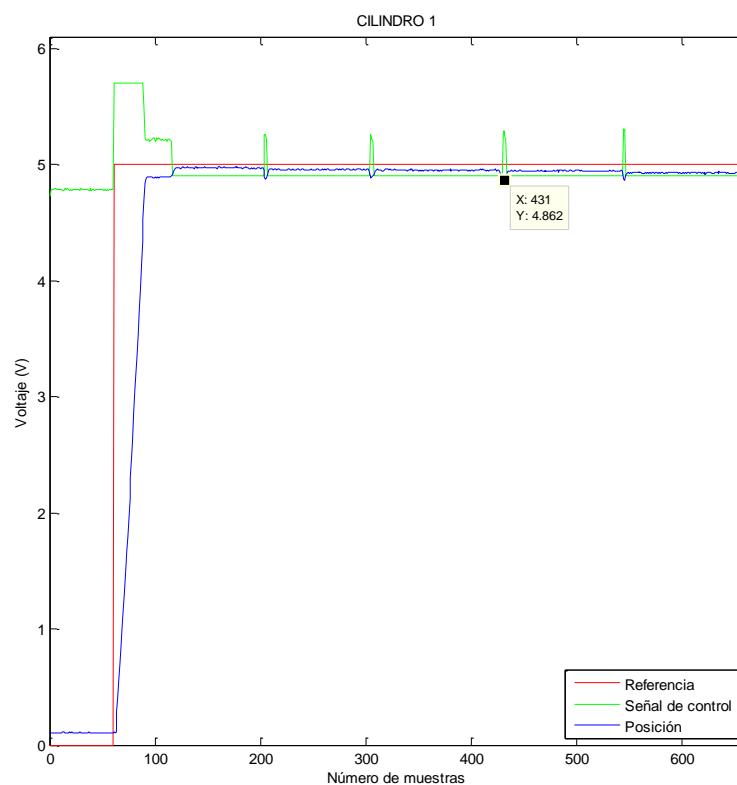
- **Perturbación del cilindro 2 debido al desplazamiento del cilindro 1.**





- Perturbación del cilindro 1 debido al desplazamiento del cilindro 2.





Los resultados conseguidos se presentan en la tabla siguiente:

	Perturbación máxima (%) Antes del ajuste	Perturbación máxima (%) Después del ajuste	Reducción conseguida (%)
CILINDRO 1	5.83	2.76	52.7
CILINDRO 2	6.70	3.78	43.6

6.3. Sintonización de los controladores PID para el movimiento simultáneo.

Para reducir las perturbaciones en la medida de lo posible se tuvieron que volver a realizar numerosas pruebas para sintonizar unos controladores PID adecuados para cada cilindro teniendo en cuenta el comportamiento de cada uno de ellos.

Los valores utilizados anteriormente provocaban la inestabilidad del sistema con bastante asiduidad. Bastaba con indicarle a ambos cilindros referencias altas y distintas para que el sistema se volviera incontrolable.

Se decidió entonces reducir la acción integral ya que, aunque no es la única que provoca inestabilidad, es la que afectaba en menor medida a la respuesta del sistema. La acción proporcional no se podía disminuir para evitar errores de posición importantes y conseguir que los cilindros respondieran a una velocidad aceptable para los propósitos de este proyecto.

Para corregir el mayor error que presentaba el cilindro derecho se le aumentó esta ganancia proporcional a su controlador. Del mismo modo, se incrementó ligeramente su acción derivativa para tratar de reducir las oscilaciones que en él tenían lugar debido a las perturbaciones causadas por el cilindro izquierdo, que como ya se ha visto, cobraban mayor importancia que en el caso opuesto.

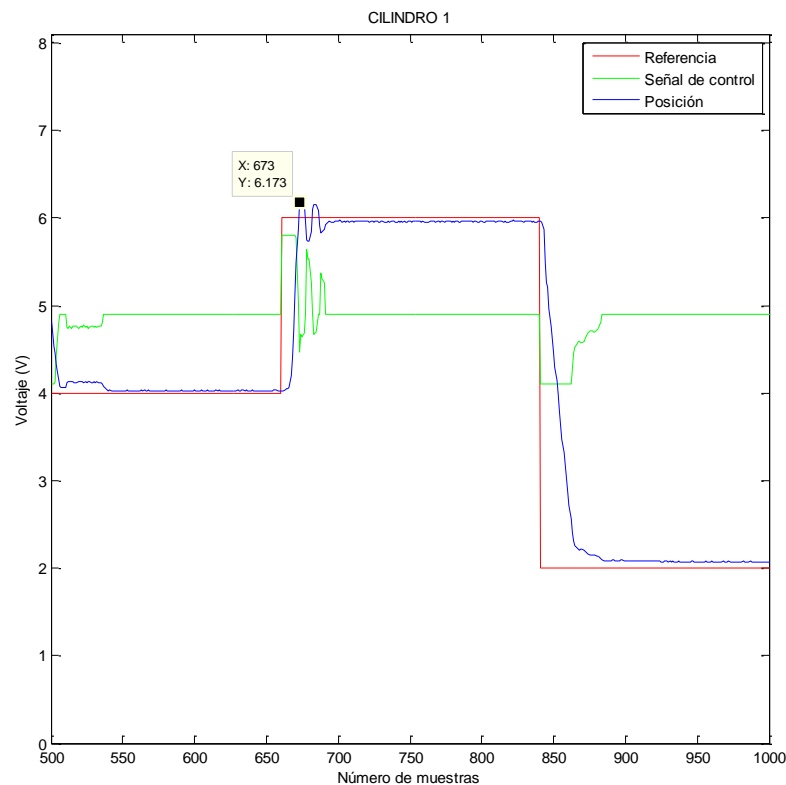


Fig. 6.2. Oscilaciones en el cilindro izquierdo

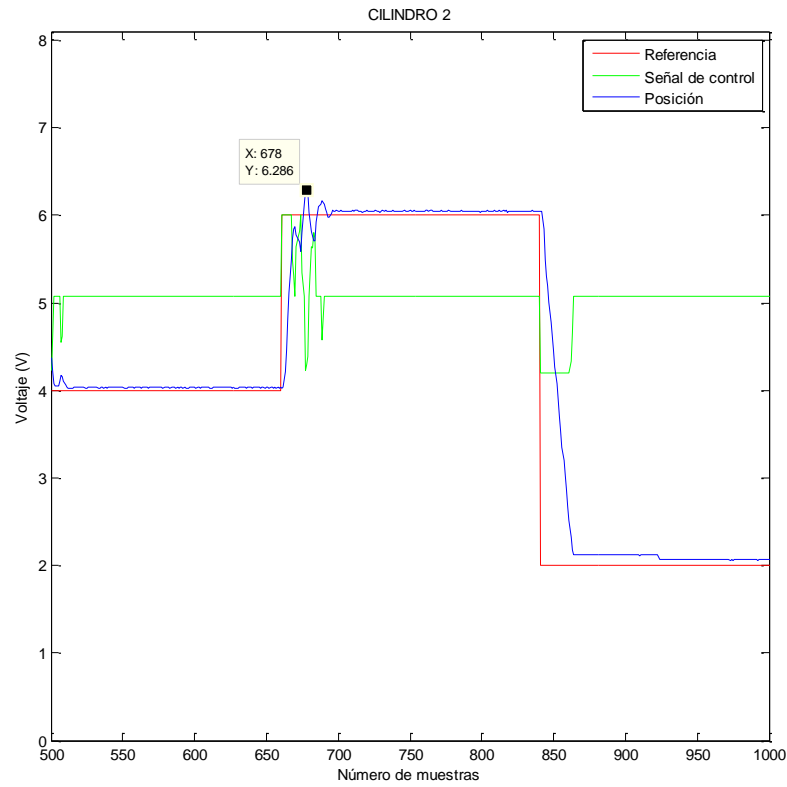


Fig. 6.3. Oscilaciones en el cilindro derecho

Los parámetros de los controladores que mejor respuesta ofrecía para los cilindros trabajando de forma simultánea se recogen en la siguiente tabla:

PARÁMETROS PID	CILINDRO 1	CILINDRO 2
Kp	2.0	2.5
Ki	0.001	0.001
Kd	0.02	0.025

Y la respuesta que estos ofrecían ante una referencia de pulsos aleatorios se presenta a continuación.

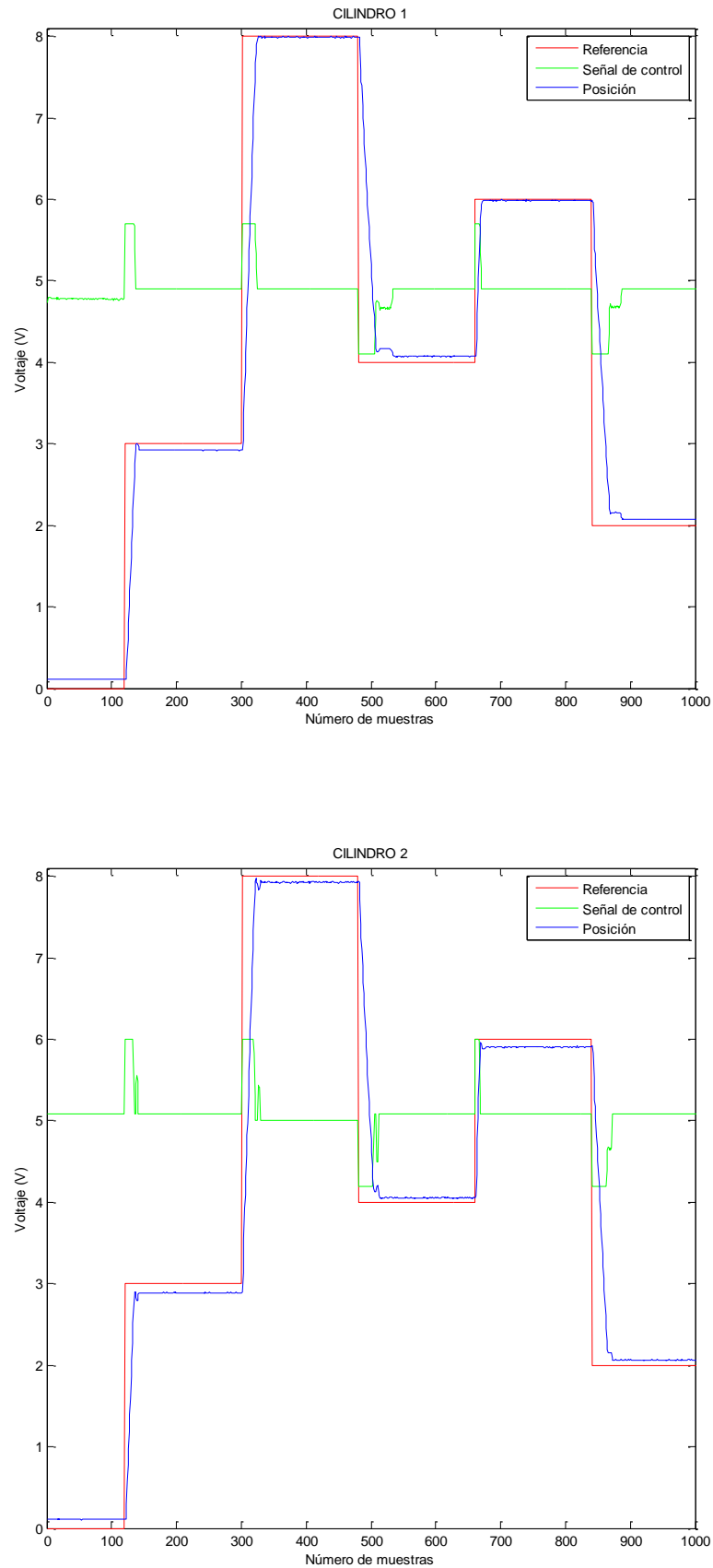


Fig. 6.4. Respuesta de ambos cilindros tras sintonizar los parámetros de los controladores

6.4. Sincronización y saturación del caudal de aire en las válvulas.

Cuando se iba a implementar el movimiento simultáneo de los cilindros se observó que la rigidez estructural de cada uno era distinta y que ambas válvulas parecían no responder con la misma precisión y velocidad, por lo que era de esperar que el desplazamiento de los vástagos fuera, en cierta forma, caótico. Efectivamente, bastaron unas pocas pruebas para corroborar este hecho: el cilindro izquierdo llegaba antes que el derecho a las referencias indicadas, es decir, iba con mayor velocidad. Sin embargo, esto solo ocurría al desplazarse de una referencia inferior a una superior. Para el caso opuesto era el cilindro derecho el que llegaba antes a su destino, aunque con una diferencia en tiempo mucho menor.

Para ajustar la velocidad de cada cilindro la solución más factible era saturar el caudal de aire en cada válvula de forma manual e independiente.

Para ello se llevó a cabo un sencillo experimento en el cual se programaba en ambos cilindros varias referencias, siendo éstas equidistantes a ellos. Cada vez que se realizaban estos desplazamientos, subidas y bajadas, se iba variando ligeramente la señal de control máxima y mínima disponible para cada válvula, es decir, se iba saturando el caudal de aire.

Finalmente las señales de control de cada válvula proporcional se comprenden entre los siguientes valores:

- CILINDRO 1: $4.1 < U(1) < 5.7$

```
%Saturación de la señal de control.  
if U(1)>5.7  
    U(1)=5.7;  
elseif U(1)<4.1  
    U(1)=4.1;  
end
```

- CILINDRO 2: $4.2 < U(2) < 6.0$

```
%Saturación de la señal de control.  
if U(2)>6  
    U(2)=6;  
elseif U(2)<4.2  
    U(2)=4.2;  
end
```

Con estos valores se consiguieron los resultados mostrados en los apartados anteriores, lo que supuso un excelente punto de partida para poder realizar seguimientos de trayectorias con el robot.

CAPÍTULO 7:
SEGUIMIENTO DE TRAYECTORIAS.

7.1. Introducción.

Mediante los procedimientos desarrollados en los capítulos anteriores se ha conseguido obtener un comportamiento de los cilindros suave, sin oscilaciones no deseadas, y bastante ajustado a las referencias introducidas. Ahora se tratará de utilizar estas propiedades para llevar a cabo un seguimiento de trayectorias lo más acertado posible.

Estas trayectorias podrían ser de cualquier tipo, pero en este proyecto nos hemos centrado en polígonos regulares y aproximaciones a curvas sencillas. Al dibujar algunas figuras ha sido conveniente modificar ligeramente los parámetros de los controladores para obtener una respuesta más rápida. No obstante, nunca se ha requerido un cambio sustancial de los mismos.

Todas las operaciones han sido realizadas a una presión de aire comprimido de 5 bar y con una frecuencia de muestreo de 30 muestras por segundo.

7.2. Tratamiento de coordenadas y espacio de trabajo.

Para llevar a cabo un seguimiento de trayectorias por coordenadas es imprescindible implementar en el programa de control la cinemática inversa. De esta forma a cada coordenada que se introduzca en el programa le corresponderá una pareja de coordenadas articulares, que son los alargamientos de los vástagos de cada uno de los cilindros. Estos valores estarán en unidades de longitud, de 0 a 30 cm. Cambiando a la escala de 0 a 10 voltios con la que operan los convertidores se tendrán los valores de referencia para la señal de control.

7.2.1. Tratamiento de coordenadas.

Las coordenadas de referencia se almacenarán en dos vectores, X e Y . Éstos, a su vez, se guardarán en dos vectores de mayor longitud de forma que cada par de valores tenga una duración determinada.

Si esta duración (que es en realidad un periodo de muestreo para las coordenadas) es demasiado pequeña, el sistema no alcanzaría las referencias. Durante el tiempo que tarden los cilindros en alcanzar la posición especificada la señal de control cambiaría para ir a la siguiente posición.

Por el contrario, si este periodo de muestreo es demasiado elevado el sistema tardará mucho en completar el trazado.

A continuación se muestra el código de tratamiento de coordenadas.

```
inimues=1;
finmues=100;
INC=1;
per=30;           %periodo de muestreo de cada par de coordenadas
for i=1:length(X)

    %X,Y: vectores de coordenadas

    EJEX(1,inimues:finmues)=X(1, INC);
    EJEY(1,inimues:finmues)=Y(1, INC);

    if inimues==1
        inimues=inimues+finmues;
    else inimues=inimues+per;
    end
    finmues=finmues+per;
    INC=INC+1;
end
```

Con este programa se consiguen dos vectores, *EJEX* y *EJEY* en los que la primera coordenada se mantiene durante 100 unidades (ya que los cilindros tardan más en alcanzar la primera referencia al partir desde su posición más retraída) y el resto durante 30 muestras, suficiente para llevar a cabo la mayoría de figuras con una buena relación velocidad – precisión.

7.2.2. Espacio de trabajo.

Para que el sistema funcione correctamente las coordenadas introducidas deben estar contenidas en el espacio de trabajo del robot.

Este espacio de trabajo se obtuvo de forma gráfica en el proyecto de *Modelado, simulación y diseño de un robot paralelo de 2 grados de libertad con acción neumática* realizado por Miguel Torregrosa Puig. Mediante cálculos en *Matlab* se llegó a las siguientes conclusiones para la estructura actual:

- Ángulo fijo = 120° (entre cilindro izquierdo y eslabón de unión).
- Distancia entre apoyos, $a = 60$ cm.
- Longitud del eslabón de unión, $b = 20$ cm.

La característica más importante de esta configuración es que permite incluir por completo un papel normalizado DIN A-4.

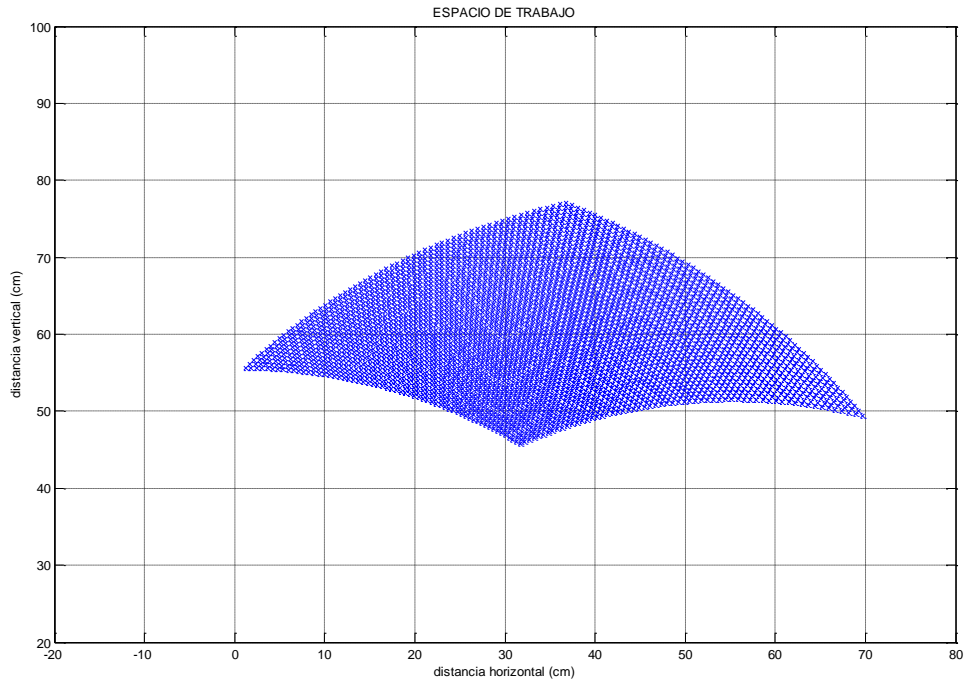


Fig. 7.1. Espacio de trabajo del robot

Como norma general las coordenadas de todas las figuras realizadas estarán comprendidas entre los siguientes valores:

	Mínimo	Máximo
Coordenadas eje X	20 cm	50 cm
Coordenadas eje Y	50 cm	70 cm

Además, en ningún caso se implementarán figuras excesivamente grandes, con lo que se asegura que el sistema no se salga de su espacio de trabajo.

7.3. Figuras realizadas.

Al cilindro horizontal actuador, el *DNC 32-100-PPV-A-S2*, se le acopló un lápiz para dibujar sobre un folio la trayectoria seguida por el robot. Este acoplamiento no estuvo muy logrado debido a la falta de tiempo y recursos pero permitió conseguir unos resultados satisfactorios.

Para todas las figuras expuestas a continuación se incluirán dos gráficas correspondientes a la posición de los cilindros en cada instante, ya que esto permite entender algunas de las imprecisiones en las trayectorias gráficas.

7.3.1. Cuadrado por coordenadas manuales.

Estas primeras pruebas consistían en introducir en el programa de control las coordenadas de un cuadrado, del siguiente modo:

```
X=[30, (30:0.5:35), 35*ones(1,11), (35:-0.5:30), 30*ones(1,11)];  
Y=[50, 50*ones(1,11), (50:0.5:55), 55*ones(1,11), (55:-0.5:50)];
```

Estos valores representan un cuadrado de 5 cm de lado. Si únicamente se indicaran los valores de los vértices los lados del cuadrado no quedarían bien definidos. Por este motivo se introduce un punto cada medio centímetro, para conseguir un buen seguimiento de los lados sin perjudicar en exceso la rapidez del proceso.

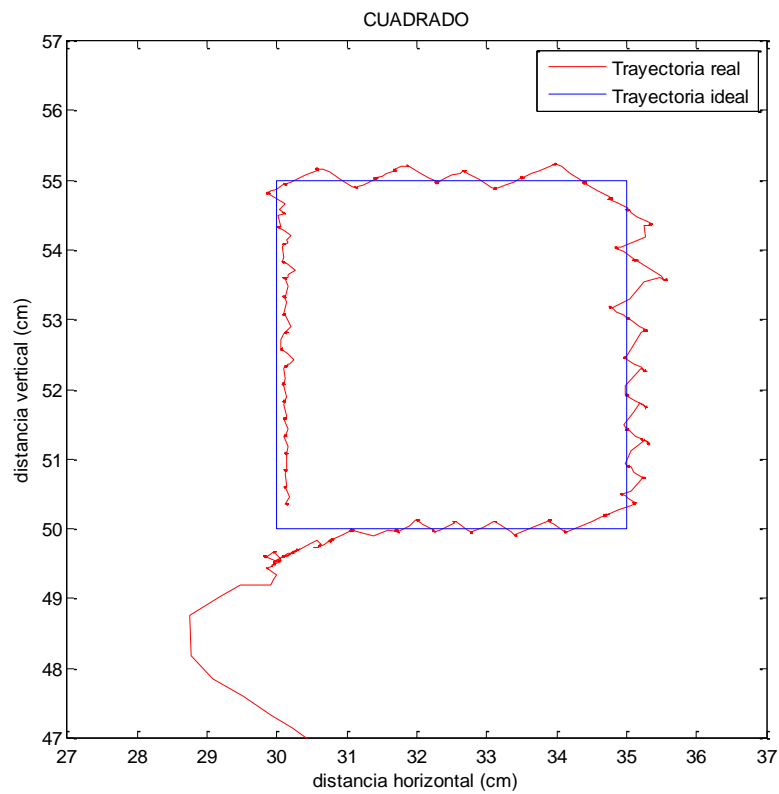


Fig. 7.2. Cuadrado: primeras pruebas

En esta primera representación se observa un importante error de posición en las esquinas de la derecha, así como en el punto inicial y final del recorrido.

En las gráficas de la posición de cada cilindro respecto a la referencia que deberían seguir se observa que los saltos de referencia que han de acometer son demasiado pequeños, inferiores a la zona muerta de los actuadores. Las válvulas no responden adecuadamente y los vástagos no se mueven hasta que el nivel de tensión ha variado lo suficiente.

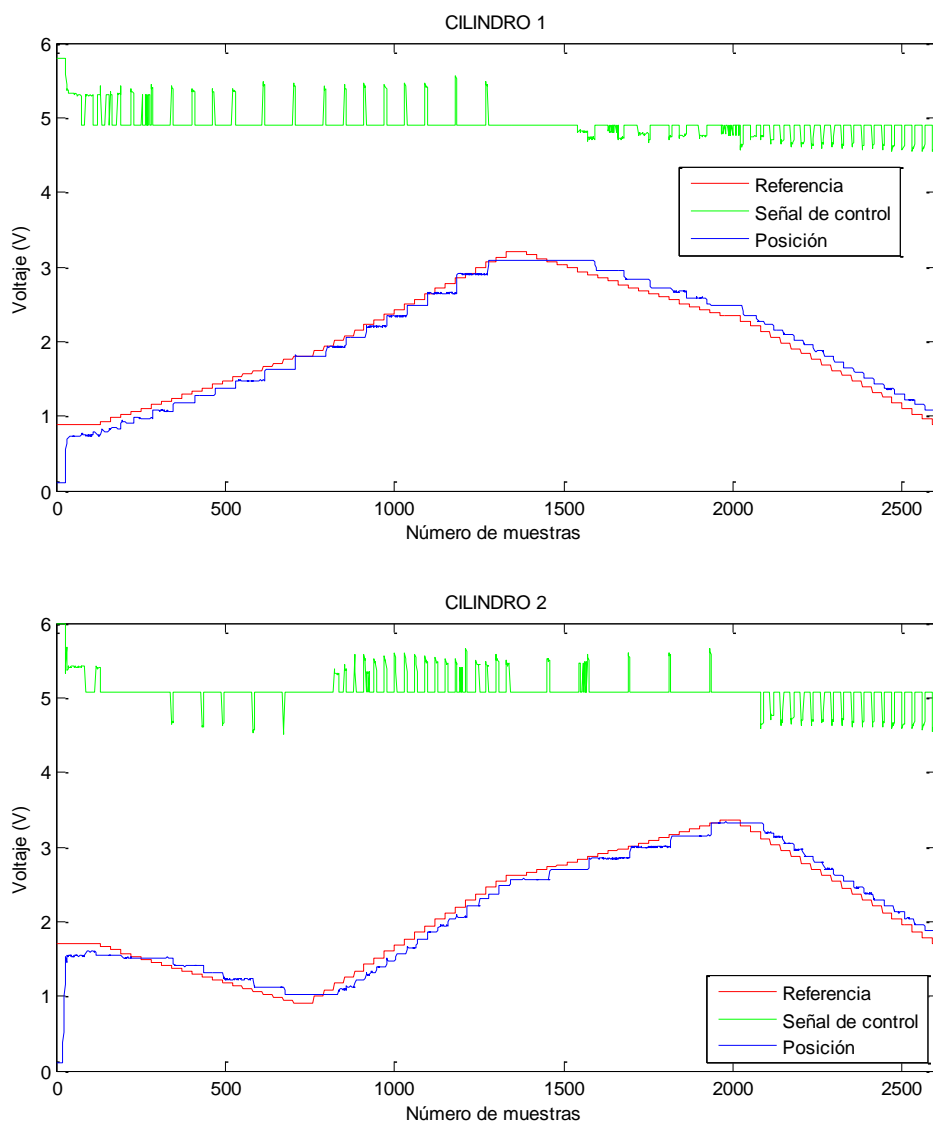


Fig. 7.3. Posición de los cilindros durante la representación del cuadrado

Esto es debido, entre otras cosas, a que el cuadrado es demasiado pequeño, lo que provoca unos cambios en la tensión que llega a las válvulas que éstas son incapaces de seguir.

Por ello, para representar de nuevo esta figura se aumentaron sus dimensiones a un lado de 10 cm y se modificaron los parámetros de los controladores de la siguiente forma:

PARÁMETROS PID	CILINDRO 1	CILINDRO 2
Kp	2.2	2.8
Ki	0.04	0.025
Kd	0.025	0.02

Estos parámetros fueron restablecidos a la configuración obtenida en el capítulo anterior tras la realización de esta prueba.

Los lados horizontales de la figura quedan bien definidos, pero en los lados verticales se dan fenómenos oscilatorios. Esto se debe al aumento de los parámetros en los controladores, que si bien permiten que los cilindros pasen por todos los puntos indicados, provocan una sobreoscilación que se refleja en el camino trazado entre cada punto de la figura.

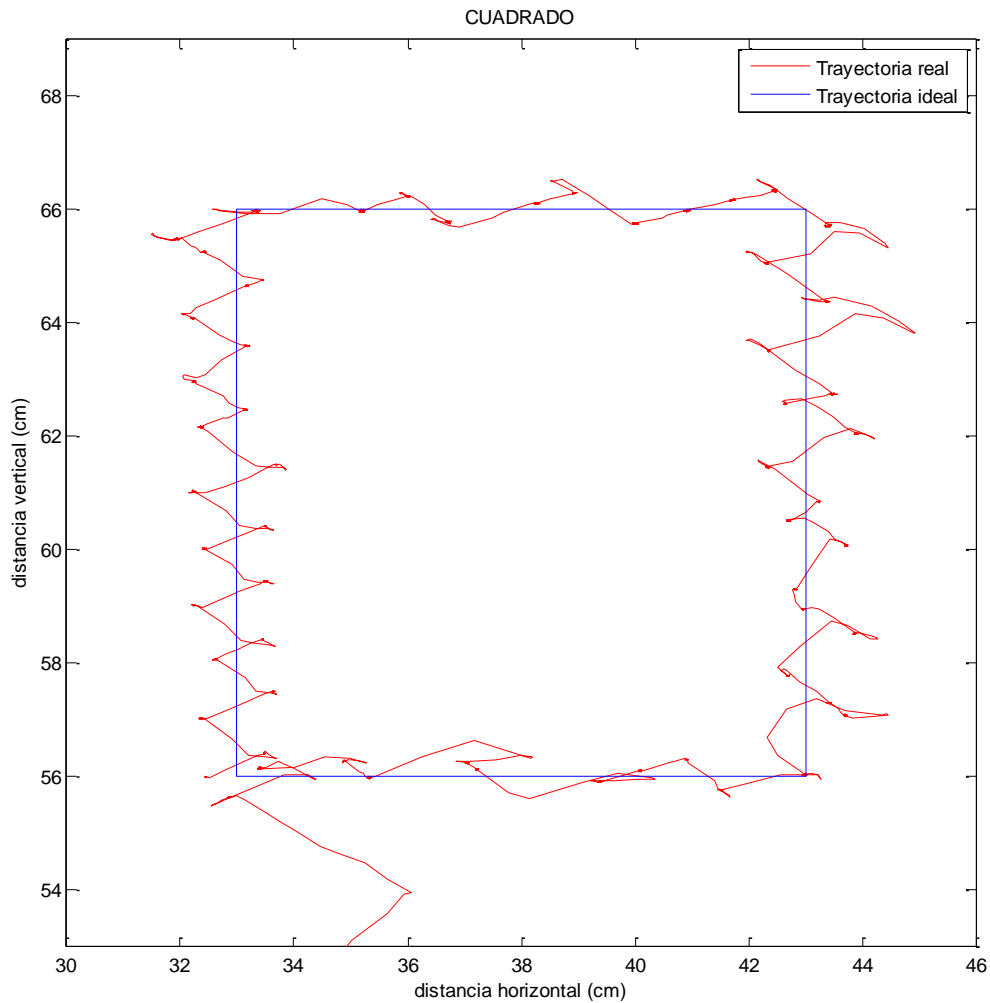


Fig. 7.4. Cuadrado tras modificar los parámetros de los controladores

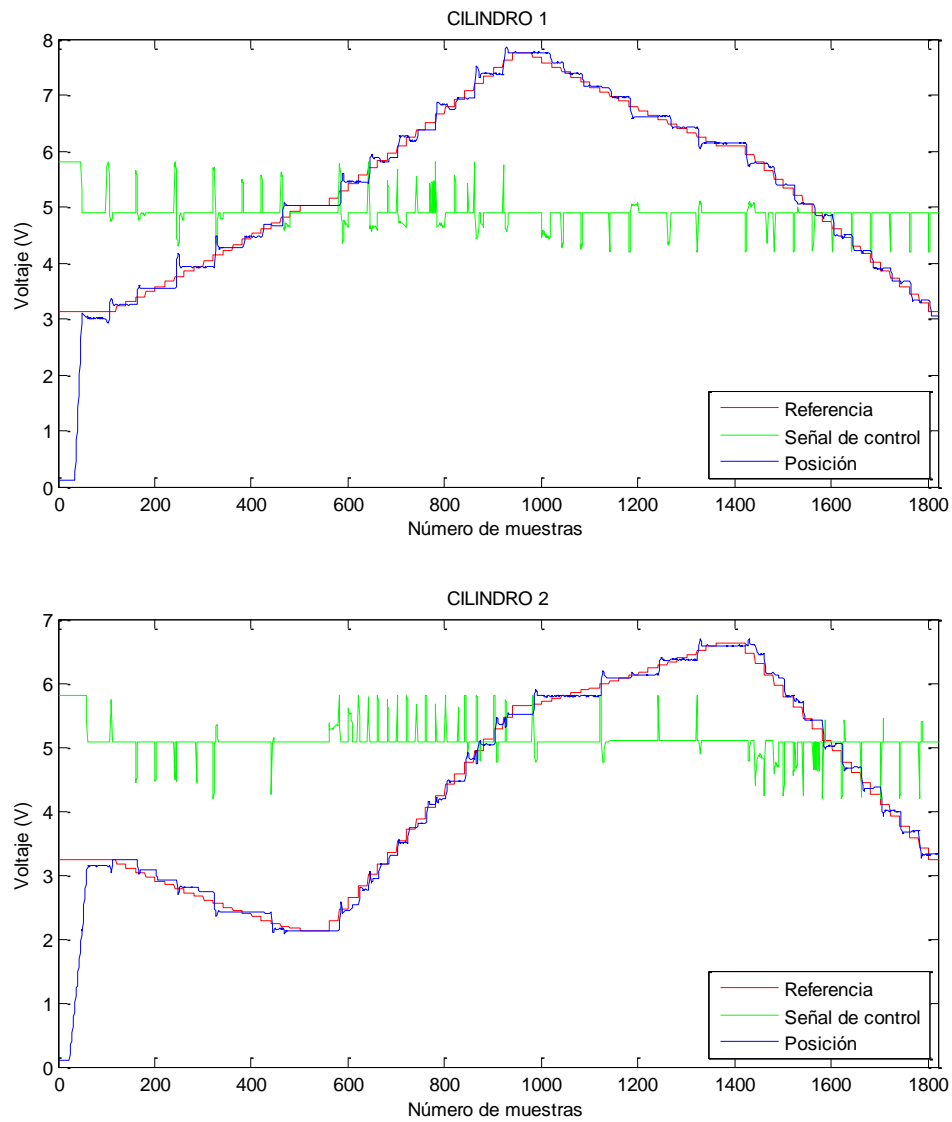


Fig. 7.5. Posición de los cilindros durante la representación del cuadrado

En estas gráficas se observa lo comentado, unas oscilaciones no deseadas en los movimientos verticales. Se comprobó que en estos desplazamientos el sistema oscilaba, ya que cada cilindro debía ir en el mismo sentido pero abarcar longitudes distintas, y los cilindros no se movían al unísono.

7.3.2. Polígonos regulares.

Las siguientes pruebas realizadas tenían como fundamento representar el contorno de cualquier polígono y, por último, un círculo de la forma más aproximada posible.

Para ello se necesitó implementar un programa que devolviera las coordenadas del contorno de un polígono a partir de su centro, radio y número de lados. Se añadió también una variable de forma que cada lado se pudiera dividir en varios puntos y mejorar así el seguimiento de la trayectoria de la figura.

7.3.2.1. Función *poligono.m*.

```
function [x,y]= poligono(xc,yc,r,n,N)

%Proporciona las coordenadas (x,y) para construir un polígono regular

%xc: coordenada 'x' del centro
%yc: coordenada 'y' del centro
%r: radio del polígono regular
%n: número de lados del polígono
%N: número de puntos para trazar cada lado

%inicializamos los vectores x e y con valor nulo
x=zeros(1, (N-1)*n+1);
y=zeros(1, (N-1)*n+1);

%almacena en dos vectores los vértices de la figura
for j=1:N-1:(N-1)*n+1
    ang(j)=2*pi*(j-1)/((N-1)*n);
    x(j)=xc+r*cos(ang(j));
    y(j)=yc+r*sin(ang(j));
end

%divide cada lado en N puntos (N-1 intervalos)
for j=2:1:(N-1)*n
    if x(j)==0 && y(j)==0
        x(j-1:j+N-2)=linspace(x(j-1),x(j+N-2),N);
        y(j-1:j+N-2)=linspace(y(j-1),y(j+N-2),N);
    else
        end
end
end
```

A partir de las coordenadas del centro del polígono, su radio, el número de lados y el número de puntos por lado este programa devuelve las coordenadas del contorno de la figura.

El número de puntos por lado se iba variando para cada figura para conseguir el mejor resultado posible.

Cabe destacar que el programa está basado en variaciones angulares, por lo que el primer y último punto que dibujará el robot será el que esté situado más a la derecha, ya que será el de ángulo = 0 radianes.

7.3.2.2. Pruebas con distinto número de lados.

- Cuadrado

La razón de que salga girado, se ha comentado anteriormente, es que la función *polígono.m* devuelve las coordenadas del contorno de la figura según el ángulo, de forma que en este caso devuelve los puntos cuyo ángulo es 0 , $\pi/2$, π y $3\pi/2$ respectivamente.

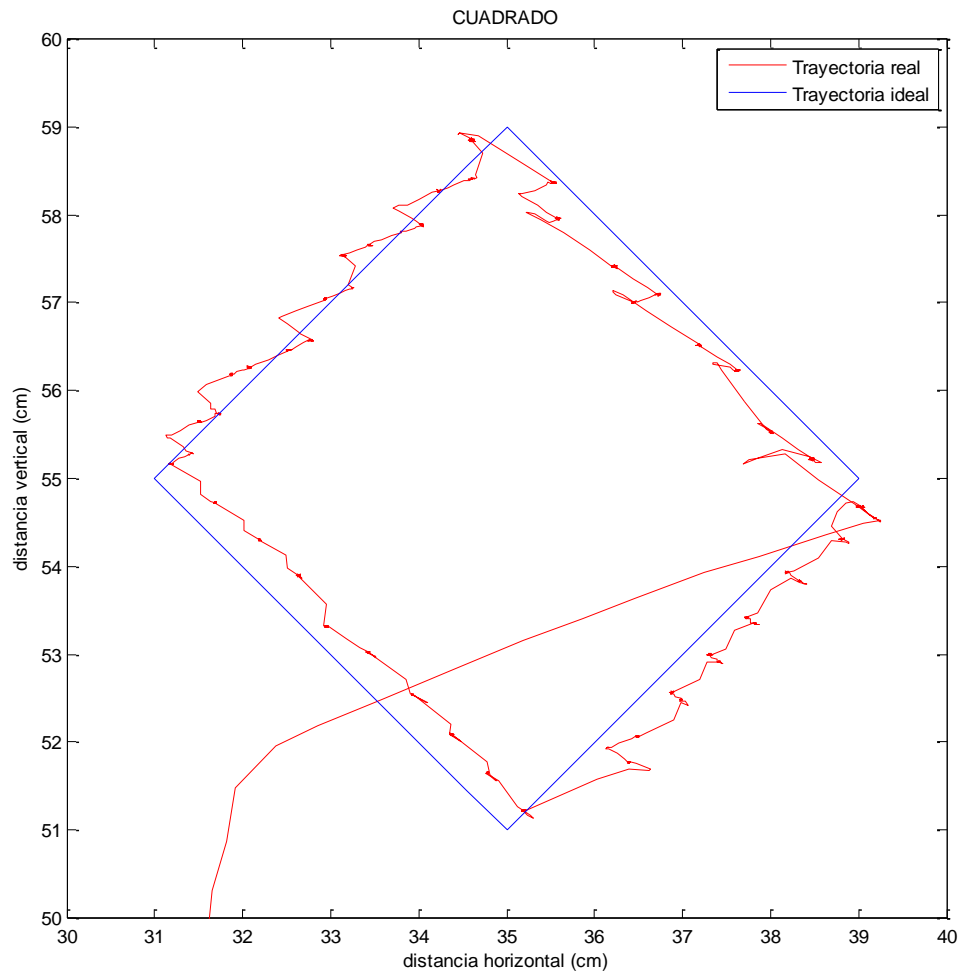


Fig. 7.6. Cuadrado mediante la función *polígono.m*

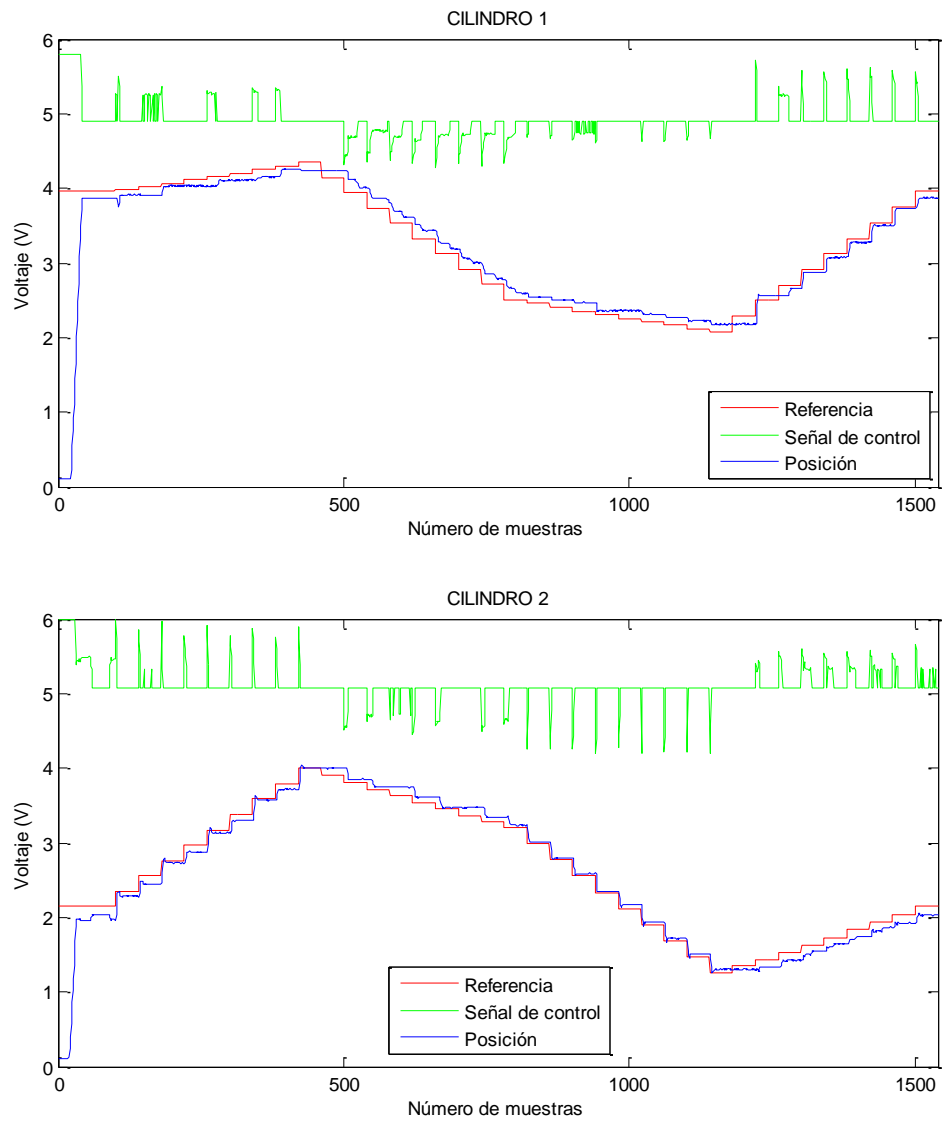


Fig. 7.7. Posición de los cilindros durante la representación del cuadrado

- Pentágono

En este caso el número de puntos por lado utilizado ha sido insuficiente, como se puede comprobar por los dos lados de la derecha en los que el cilindro actuador ha alcanzado los puntos indicados pero no sigue el camino adecuado entre ellos.

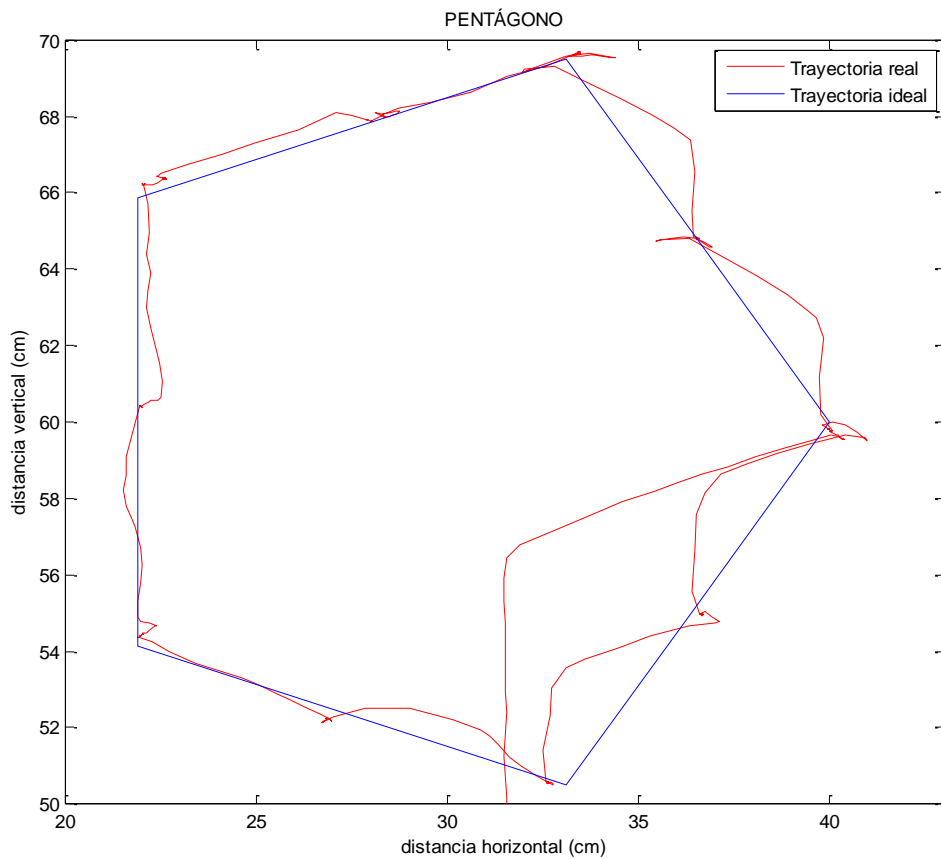


Fig. 7.8. Representación de un pentágono

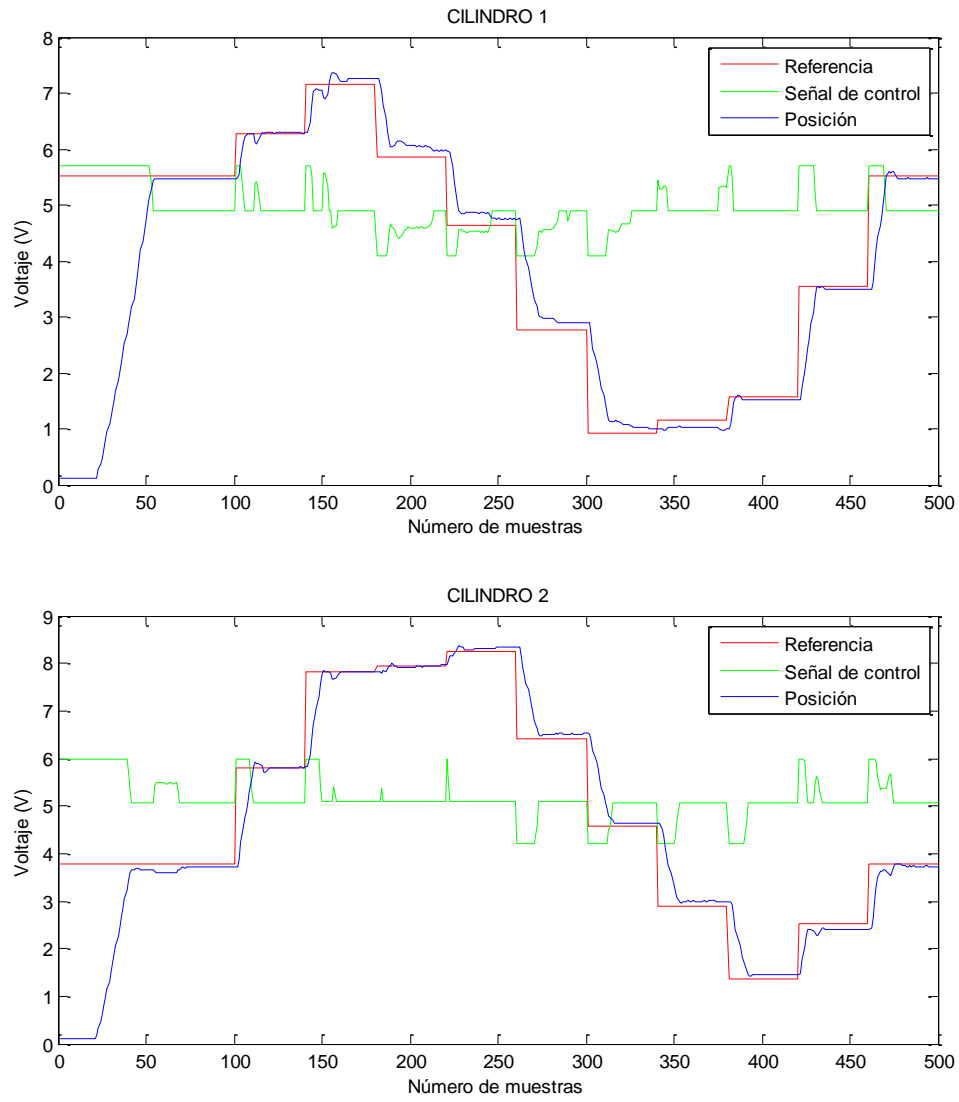


Fig. 7.8. Posición de los cilindros durante la representación del pentágono

- Hexágono

En este primer caso se utilizan tres puntos por lado dando lugar, como en la figura anterior, a errores en el camino seguido entre dos puntos consecutivos de la figura.

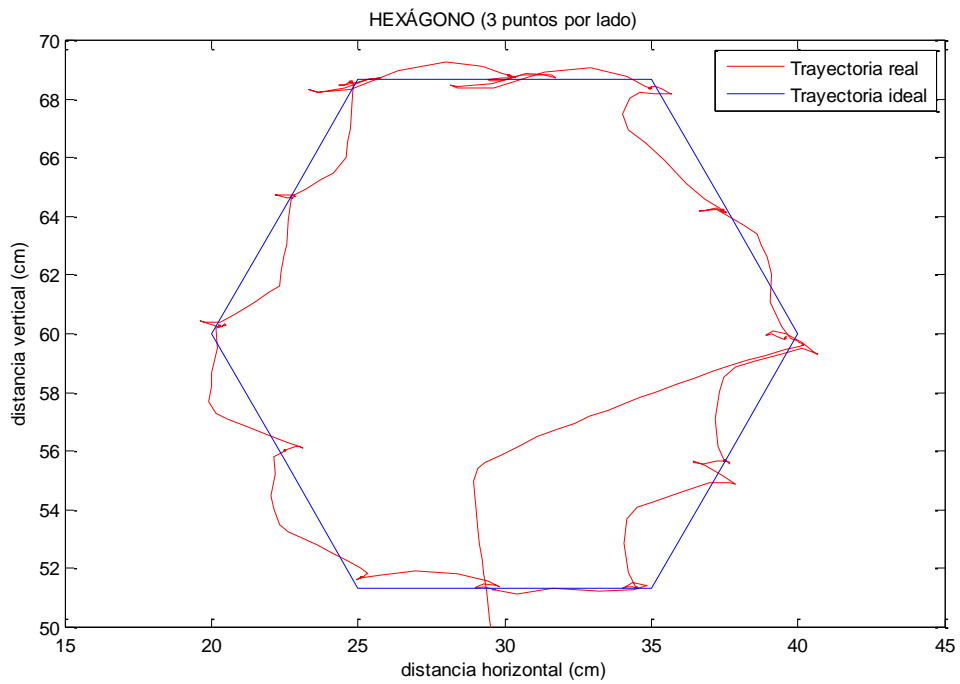


Fig. 7.9. Hexágono con tres puntos por lado

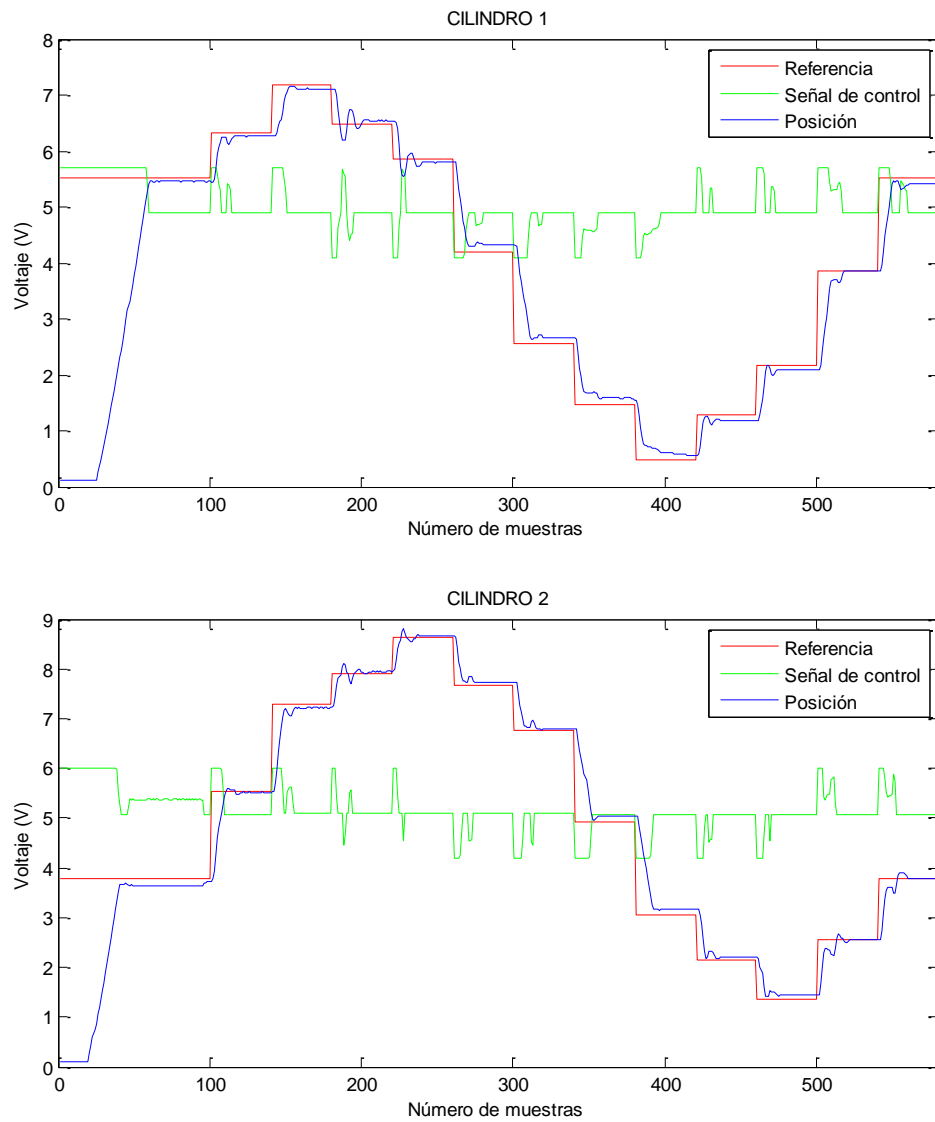


Fig. 7.10. Posición de los cilindros durante la representación del hexágono

En este segundo caso se utilizan cuatro puntos por cada lado y se reducen los errores de posición del actuador en la trayectoria entre dos puntos consecutivos.

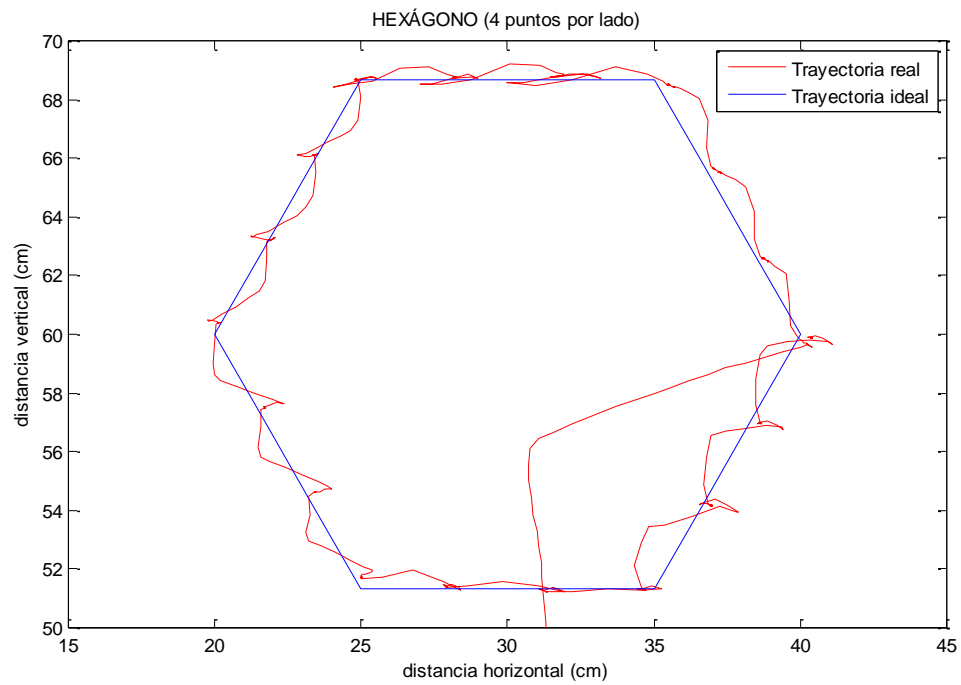
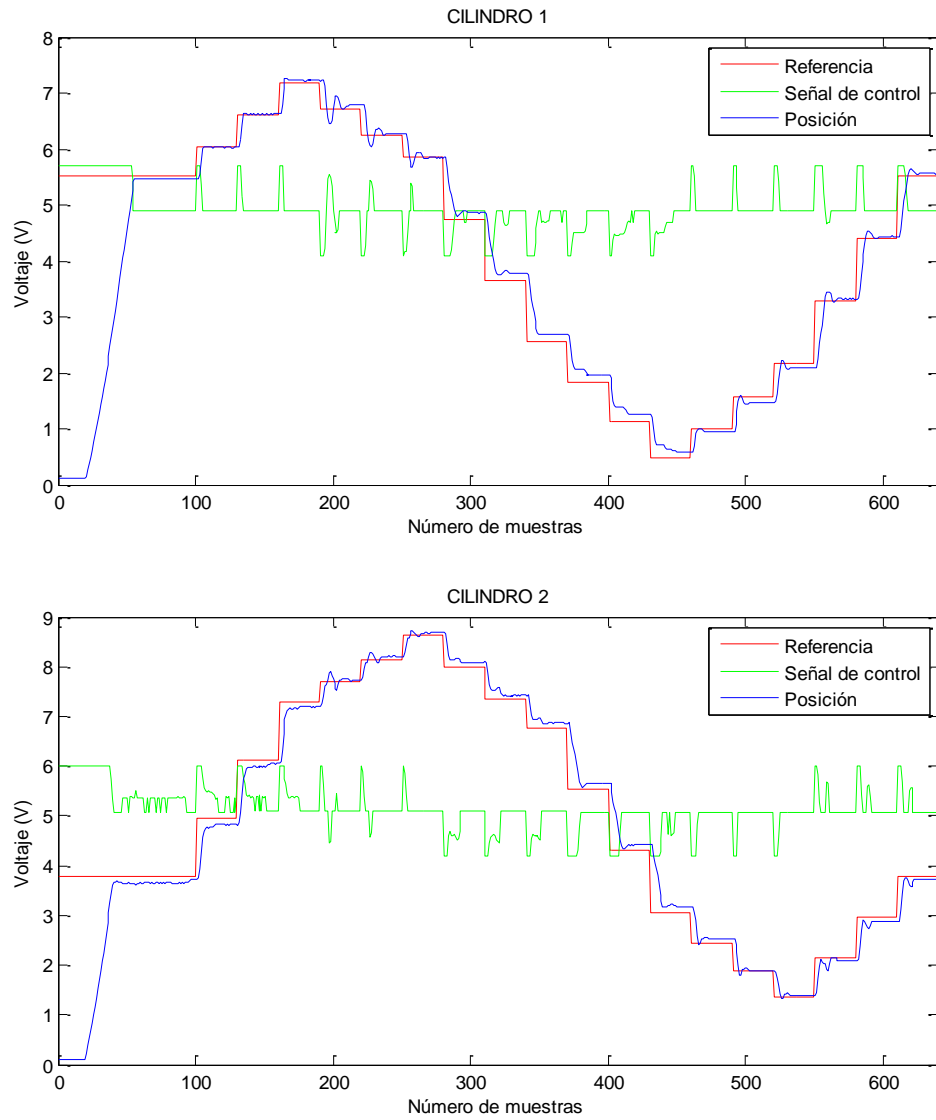


Fig. 7.11. Hexágono con cuatro puntos por lado

**Fig. 7.12. Posición de los cilindros durante la representación del hexágono**

- Octógono

En esta ocasión se optó por introducir cinco puntos por cada lado, con lo que se consiguió un muy buen seguimiento del contorno de la figura.

El número de muestras requeridas aumentó de entre 500 y 600 para las figuras anteriores hasta más de 1000, lo que supone unos 15 segundos más de tiempo total en completar la figura (recuérdese que la frecuencia de muestreo es de 30 muestras por segundo).

Este incremento en el tiempo se consideró aceptable teniendo en cuenta la notable mejora que se obtuvo en el contorno del polígono.

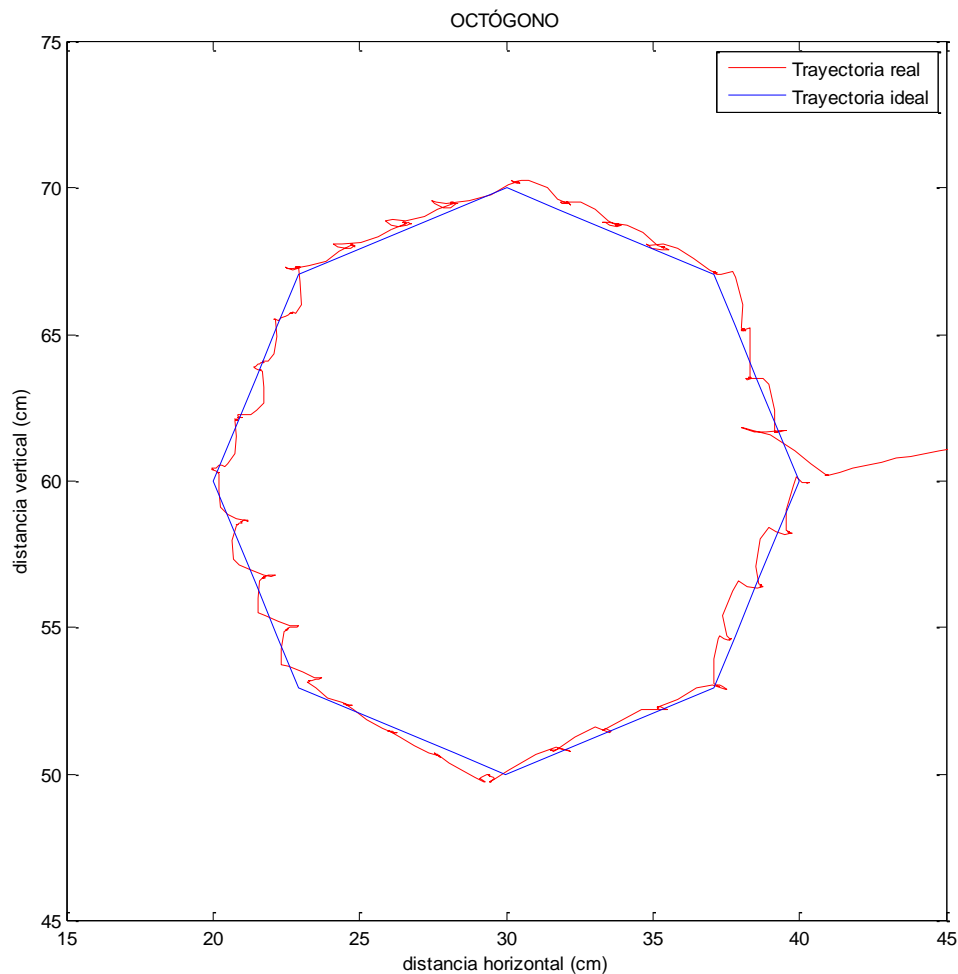


Fig. 7.13. Representación de un octógono

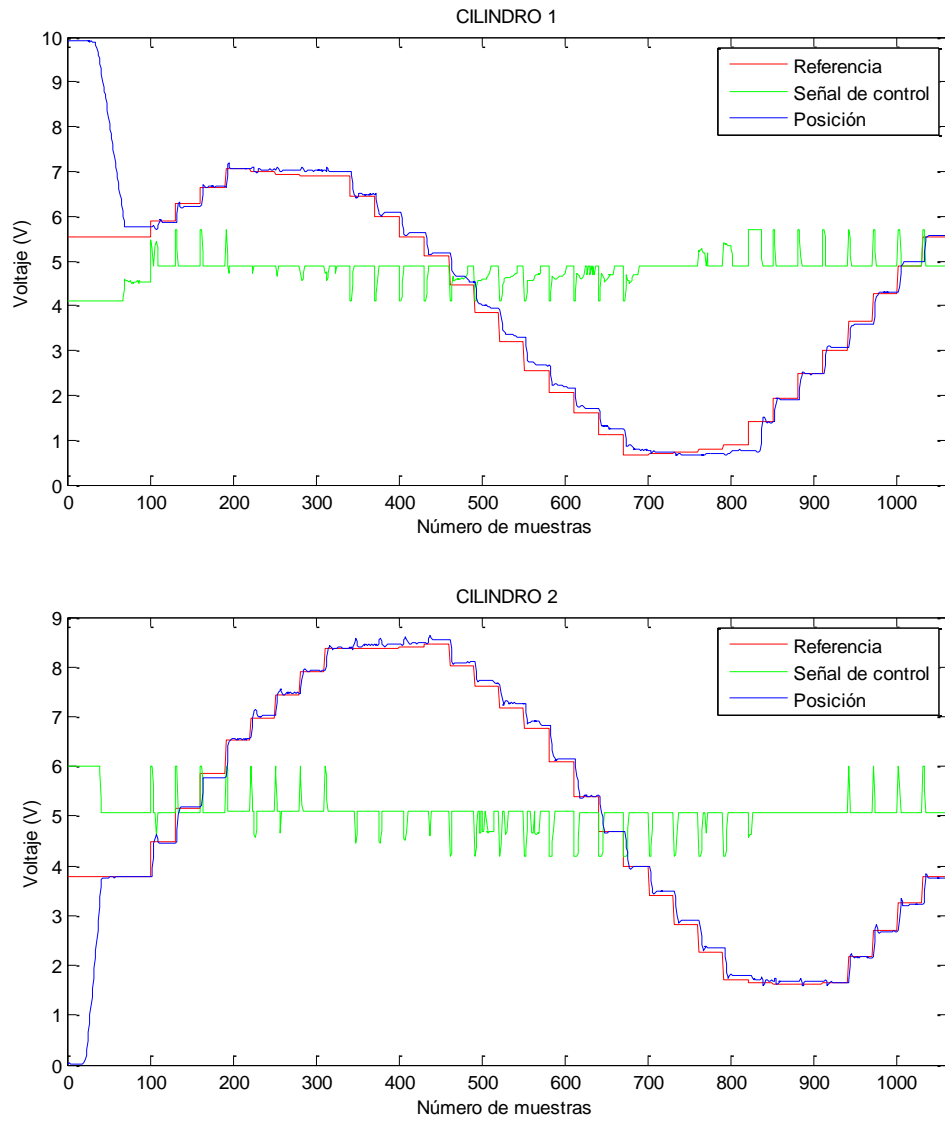


Fig. 7.14. Posición de los cilindros durante la representación del octógono

7.3.3.3. Aproximación al círculo.

El siguiente propósito consistía en conseguir realizar una circunferencia, o al menos, un polígono con un número de lados lo suficientemente elevado para que en la gráfica y en el papel se observara una circunferencia.

Tras realizar numerosas pruebas se decidió que 20 lados era un número aceptable para conseguir el contorno de un círculo de radio 10 cm. Se le indicó al programa que introdujera un punto medio en cada lado para reforzar el seguimiento del contorno.

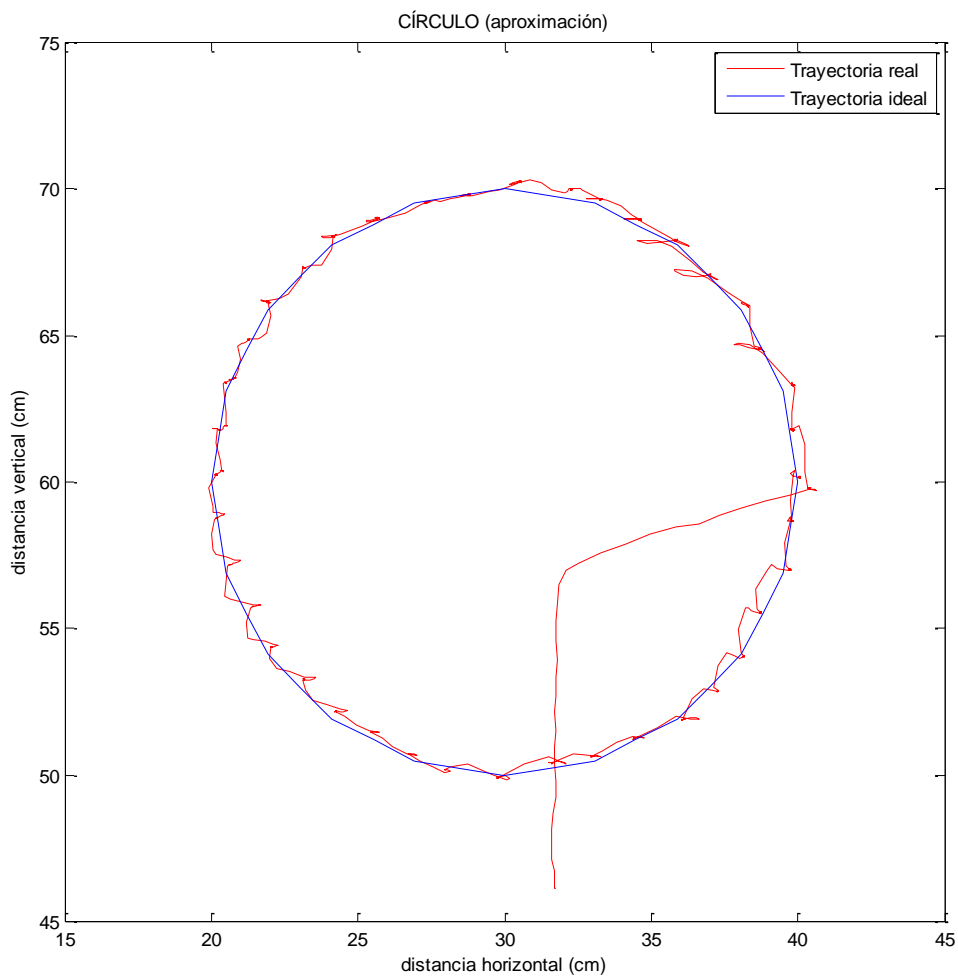


Fig. 7.15. Aproximación al círculo

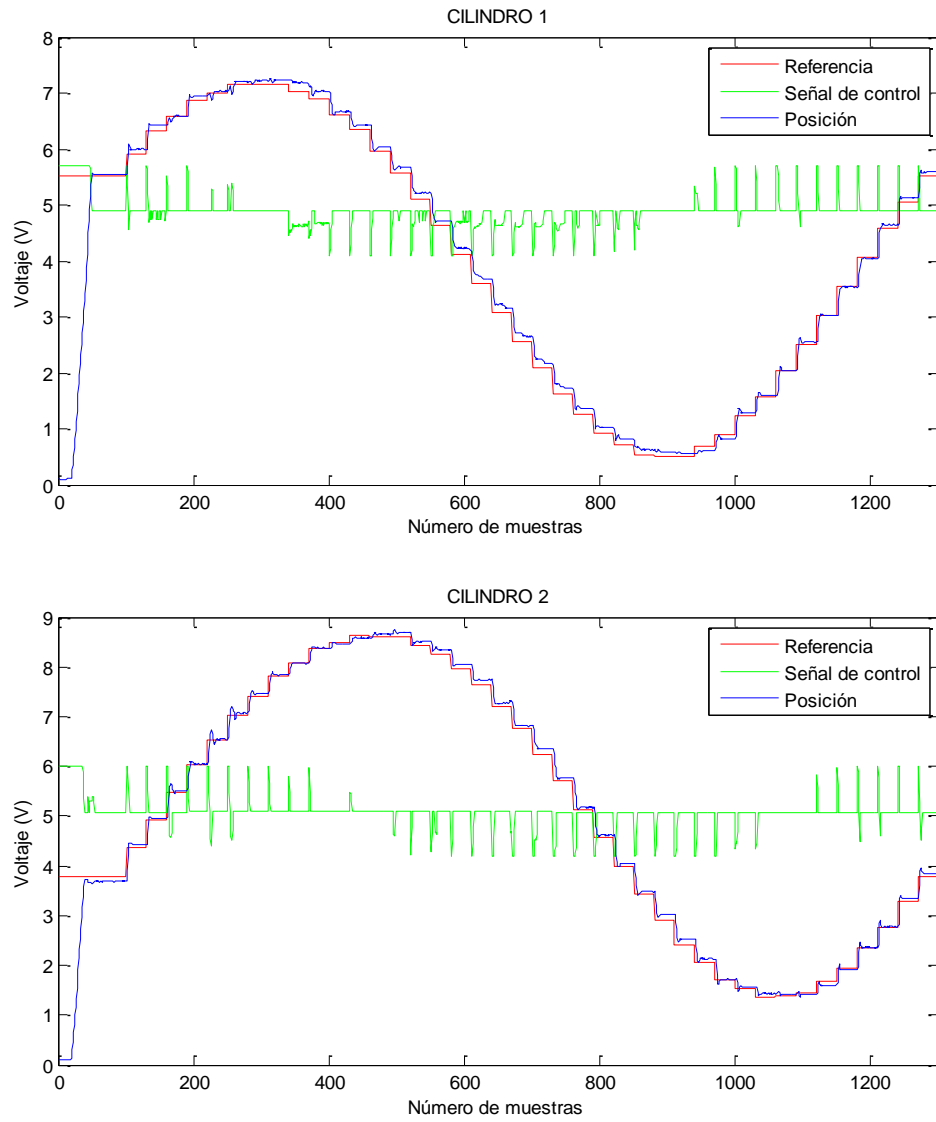


Fig. 7.16. Posición de los cilindros durante la representación del círculo

7.3.3. Senoide.

Para conseguir representar una función senoidal se implementó un programa que devolvía los puntos de su contorno en función de la amplitud indicada y una constante que se llamó *cc*.

Esta constante se introdujo para conseguir que los puntos de la senoide entraran en el espacio de trabajo ya que, de lo contrario, la figura estaría centrada en $y=0$.

Se realizaron multitud de pruebas hasta conseguir que la figura entrara toda ella en el espacio de trabajo, modificando la constante, la amplitud y las coordenadas del eje *x* sobre las que pasaría la figura.

- Senoide corta (1 periodo)

Se fijaron las coordenadas del eje *x* entre los centímetros 25 y 35 del espacio de trabajo y la amplitud en 3 cm.

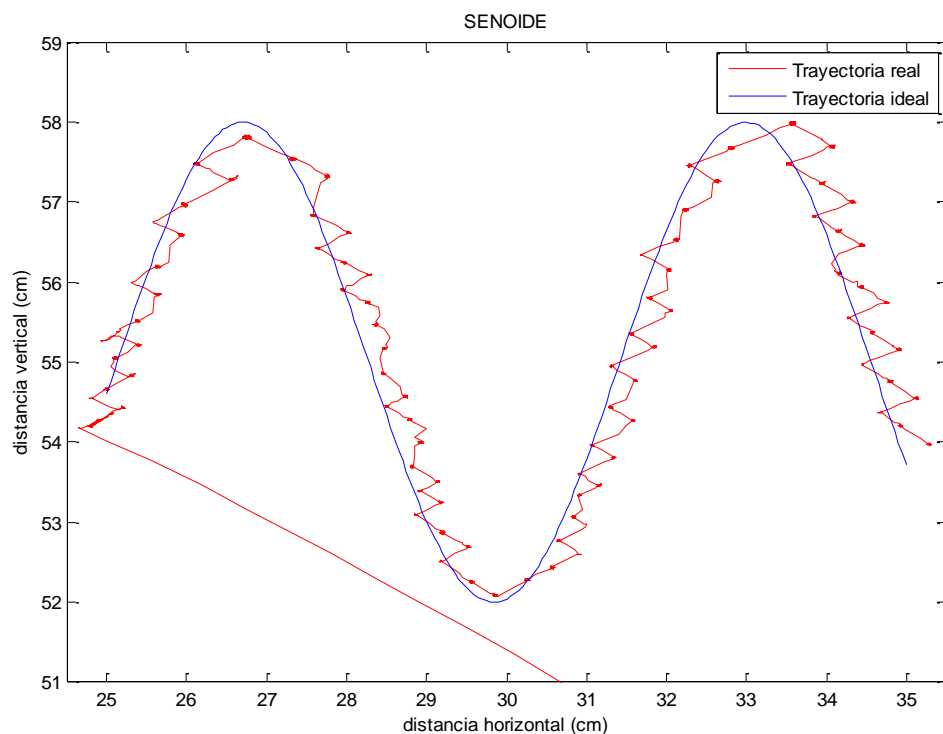


Fig. 7.17. Representación de una senoide

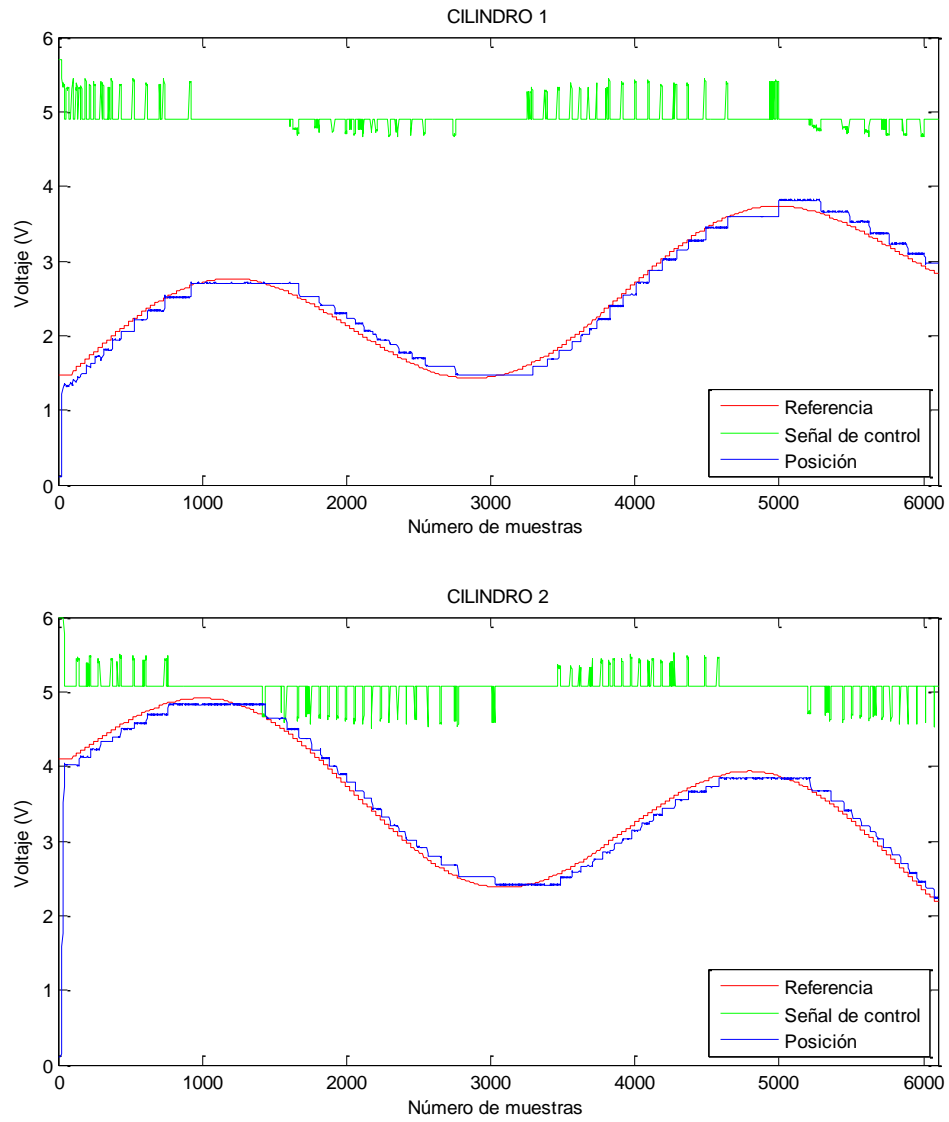


Fig. 7.18. Posición de los cilindros durante la representación de una senoide

- Senoide larga (4 periodos)

Esta vez se fijaron los valores del eje x abarcando la máxima distancia horizontal posible y se estableció la amplitud en 5 cm.

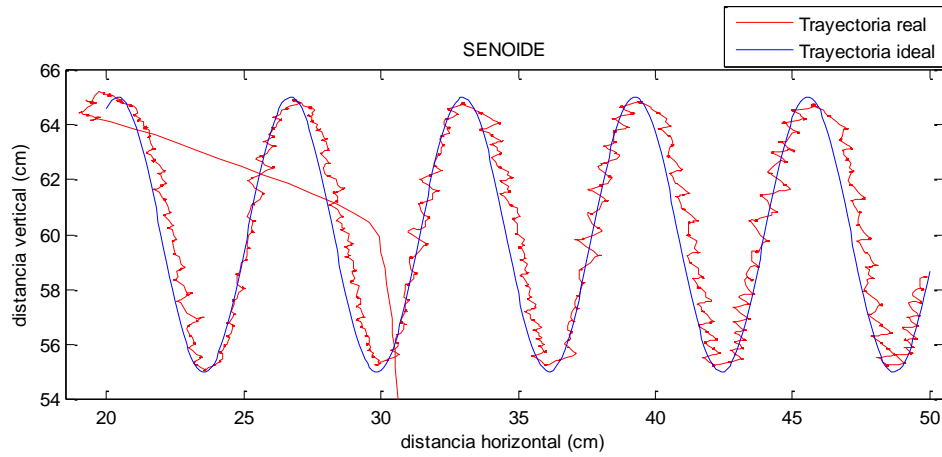


Fig. 7.19. Representación de una senoide

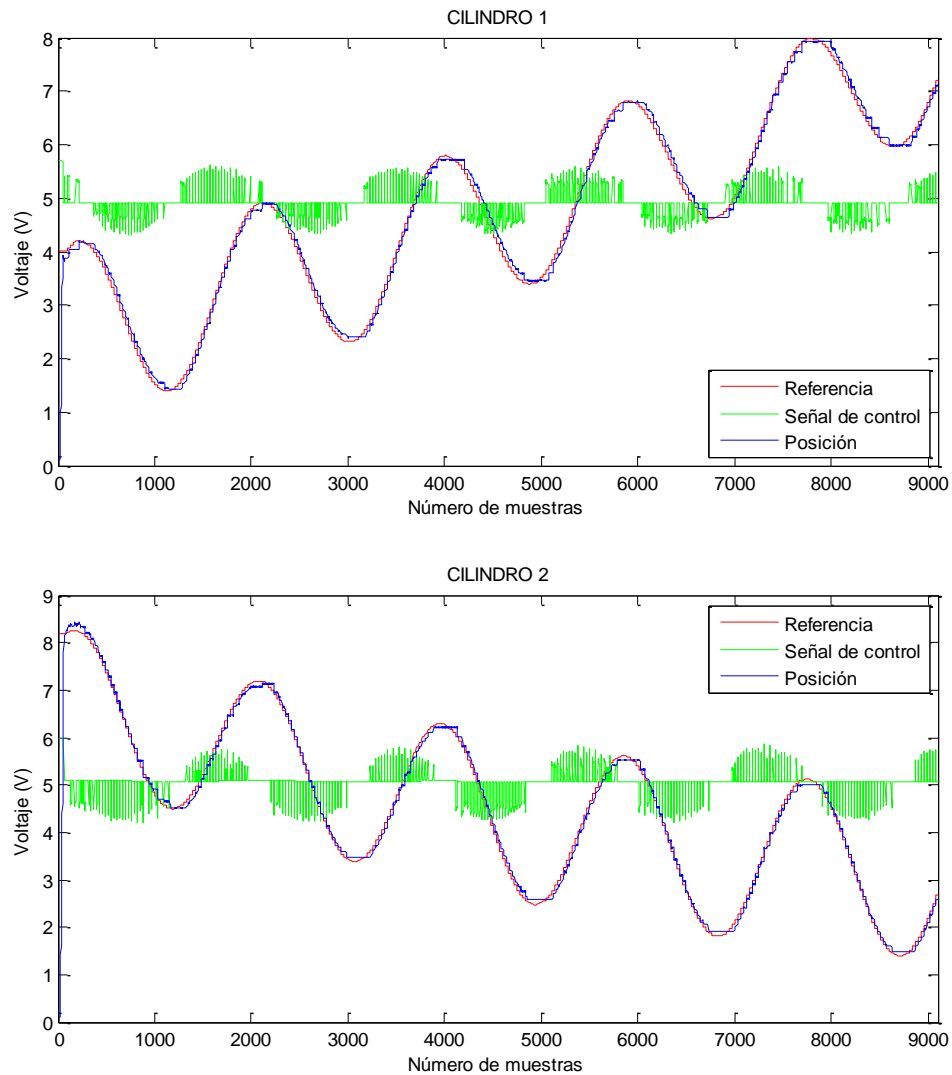


Fig. 7.20. Posición de los cilindros durante la representación de una senoide

Los resultados obtenidos para la representación gráfica de las señales senoidales tienen varios inconvenientes. En primer lugar, el tiempo necesario para su trazado es excesivamente alto (en torno a 200 segundos para el primer caso y 300 segundos para este último). Además, debido a que se trata de una trayectoria curva, los incrementos en los niveles de tensión de referencia son muy pequeños, dando lugar a la aparición del fenómeno de la zona muerta de los actuadores. Por este motivo las figuras siguen el trazado en zigzag, impidiendo la aparición de una curva suave.

No obstante, el trazado de la figura senoidal es seguido por los cilindros neumáticos con una precisión considerable que permite albergar esperanzas sobre las posibles aplicaciones prácticas del robot.

CAPÍTULO 8:
CONCLUSIONES Y DESARROLLOS
FUTUROS.

8.1. Conclusiones.

En el presente proyecto se ha continuado con el trabajo desarrollado en anteriores documentos relativos a *Robowriter*. El objetivo fundamental a perseguir era controlar la posición del extremo actuador de forma que se consiguiera realizar un seguimiento de trayectorias planas.

Para ello se han verificado los aspectos más básicos del sistema. Se revisó el conexionado eléctrico y neumático de los sensores y actuadores, la cinemática directa e inversa del robot y los programas relacionados con el motor de adquisición de datos de *Matlab*.

Mediante la extracción de una gran variedad de gráficas se obtuvieron las funciones de transferencia de cada cilindro. Sin embargo, la complejidad de la planta y su variabilidad en la respuesta ante entradas similares imposibilitó que estas funciones fueran de gran utilidad en la sintonización de controladores mediante reglas teóricas. Por este motivo se usaron métodos experimentales y con ellos se consiguieron establecer unos controladores PID que respondían perfectamente al movimiento desacoplado del sistema, con un error siempre inferior al desajuste en cero que proporcionan los propios sensores.

Al estudiar el sistema en su conjunto, con los dos cilindros acoplados, aparecían unas perturbaciones que provocaban una cierta inestabilidad. Empleando de nuevo métodos experimentales se consiguieron reducir estas perturbaciones y se implementaron unos nuevos controladores PID.

Finalmente se desarrollaron figuras geométricas y curvas sencillas para demostrar el grado de efectividad de los controladores PID en el control de posición mediante actuadores electroneumáticos. Se demostró que variando la relación precisión-velocidad del robot mediante cambios en las referencias a seguir por el efector del robot se pueden conseguir seguimientos de trayectorias planas altamente fiables.

El control ejercido por los controladores PID junto a la tarjeta de adquisición de datos, aún con todos sus defectos en cuanto a robustez y dependencia de un PC, proporciona un grado de precisión suficiente para convertir al robot en una herramienta muy versátil. Sin embargo, si se pretende conseguir niveles de precisión y maniobrabilidad más elevados conviene corregir ciertos aspectos de la estructura y componentes que se detalla en el siguiente subapartado.

En cuanto al grado de aplicación práctica, además del dibujo de figuras sobre un folio, se podrían desempeñar tareas muy diversas simplemente añadiendo la pieza adecuada al efector del robot: corte de una lámina para obtener una figura determinada, escritura sobre un teclado de ordenador, microsoldadura, punzonado, etc.

8.2. Desarrollos futuros.

Los trabajos futuros han de ir encaminados principalmente a la mejora del control de posición en términos de una mayor suavidad y velocidad en el desplazamiento de los cilindros. Para llevar a cabo tales objetivos podrían plantearse las siguientes tareas:

- Rigidización de la estructura.

La estructura del robot ha sufrido diversos accidentes a lo largo del tiempo, lo que ha provocado la aparición de ciertas holguras no deseables en la base de los cilindros neumáticos y en el efector final. Todo ello da lugar a pequeñas oscilaciones que impiden un control más preciso.

La estructura que alberga el robot también ha sido modificada debido a un error en su diseño original que obligó a retirar la placa derecha de la misma. Esto ha supuesto una mayor flexibilidad en la misma que provoca la inestabilidad de la placa posterior cuando el rotulador contacta con el folio pegado a ella.

La reparación de la estructura del robot y de su soporte sería un primer paso para obtener resultados más favorables en cuanto al control de posición.

- Revisión de actuadores y cableado eléctrico.

Durante el desarrollo del proyecto han surgido multitud de problemas derivados de un mal funcionamiento de alguno de los cilindros neumáticos, de las válvulas proporcionales conectadas a ellos y de las conexiones eléctricas. La revisión de estos componentes, y su sustitución si fuera necesario, proporcionaría al sistema un movimiento más fluido y, junto a la rigidización de la estructura del robot permitiría abordar problemas con otras soluciones distintas a las que aquí se han implementado.

Por ejemplo, se podría incrementar la presión de funcionamiento y modificar los parámetros de los controladores para obtener una respuesta más rápida sin los inconvenientes de una elevada tendencia a la inestabilidad. Además, sería posible ampliar el rango de saturación de la señal de control con el consiguiente aumento de velocidad.

- Desarrollo del control digital para el cilindro de vástago pasante.

A la hora de controlar la posición del efector final se cuenta con dos válvulas de control todo/nada. Con el fin de regular la presión que ejerce el lápiz sobre el papel se debe estudiar la frecuencia con la que estas válvulas trabajan de forma que, modificando el ancho de pulso de las señales introducidas y recibiendo los datos del transmisor de posición, el vástago se pueda desplazar de forma controlada a lo largo del rango de actuación del transmisor. En principio la frecuencia a la que estas válvulas operan es muy elevada por lo que habría que estudiar una reducción de la presión de aire de suministro o un incremento en la frecuencia de la señal digital que llega a las válvulas para poder llevar a cabo este proceso. Esto sería de gran utilidad para aplicaciones más elaboradas como la utilización de herramientas de soldadura o punzonado en las que la presión del efector sobre el plano de trabajo adquiere una mayor relevancia.

- Creación de interfaz gráfica.

Una gran mejora en el aspecto estético de los programas desarrollados para el funcionamiento del robot consistiría en programar una interfaz gráfica que pudiera usar cualquier persona para obtener los resultados del desplazamiento de los cilindros ante cualquier trayectoria realizada. Se deberían introducir unas trayectorias preestablecidas e implementarlas en un programa que pusiera en marcha el robot y devolviera las gráficas de la posición de los cilindros y el actuador.

- Aplicación de otros métodos de control.

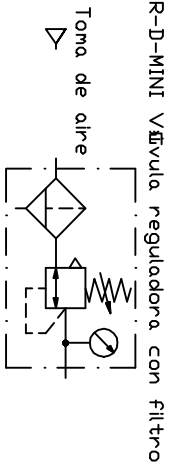
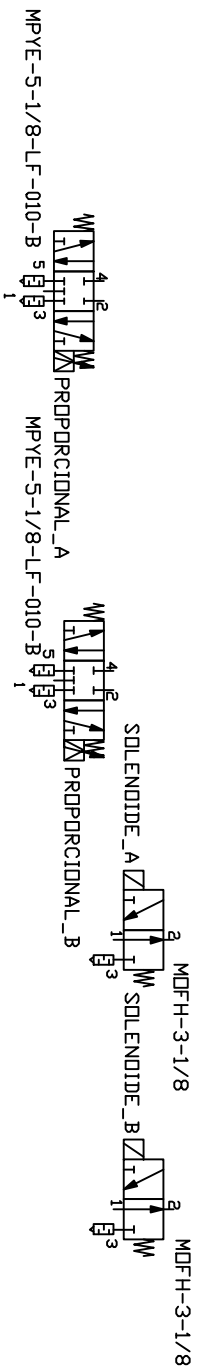
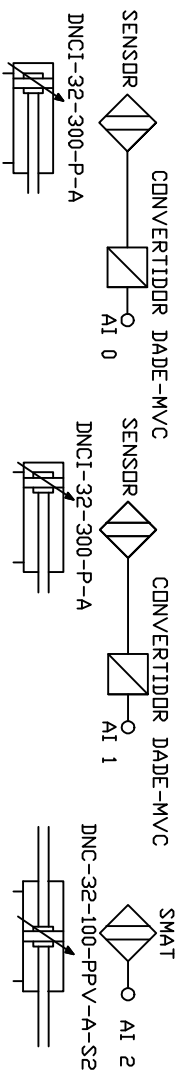
Esta opción supondría una sustitución radical de todo el conexionado eléctrico y de los programas informáticos desarrollados pero no cabe duda de que supondría una mejora en cuanto a prestaciones y versatilidad del robot.

Se pueden plantear dos alternativas, la sustitución de la tarjeta de adquisición de datos por el uso de un dispositivo PLC, o la sustitución del control mediante PID por técnicas de control adaptativo más adecuadas como el control predictivo MPC (Model Predictive Control) utilizando la propia tarjeta de adquisición de datos.

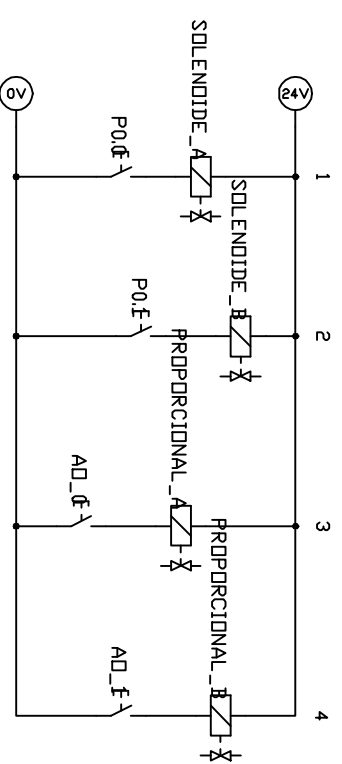
BIBLIOGRAFÍA.

- [1] Torregrosa Puig, M. (2006). *Modelado, simulación y diseño de un robot paralelo de 2-GDL con acción neumática*. Universidad Politécnica de Cartagena.
- [2] Rosillo Meseguer, A. (2007). *Montaje, programación y puesta en marcha de un robot neumático de estructura paralela*. Universidad Politécnica de Cartagena.
- [3] Moreno Martínez, G. (2009). *Control de un robot paralelo neumático de dos grados de libertad*. Universidad Politécnica de Cartagena.
- [4] Ogata, K. (1997). *Ingeniería de control moderna. 3ª Edición*. Universidad de Minnesota. Pearson Educación.
- [5] Espinosa, J. (2003). *Control lineal de sistemas multivariables*. Corporación Universitaria de Ibagué.
- [6] Astrom, K. y Hagglund, T. (1995). *PID Controllers: Theory, Design and Tuning. 2ª Edición*. Estados Unidos.
- [7] Giménez García, J. (2008). *Calibración, control y diseño SCADA de un robot paralelo neumático con el automata S7-300*. Universidad Politécnica de Cartagena.
- [8] Aracil, R., Salterén, R. J., Sabater, J. M. y Reinoso, O. (2006). Robots paralelos: Máquinas con un pasado para una robótica del futuro. *Revista Iberoamericana de Automática e Informática Industrial*, 3, 16-28.
- [9] *Historia del control automático*. <http://usuarios.multimania.es/automatica/index.htm>
- [10] Rojas Moreno, A. (2011). *Control de procesos práctico y avanzado*.
- [11] Mendiburu Díaz, H. A. (2006). *Instrumentación virtual industrial*. Perú: INDECOPI
- [12] Departamento de Electrónica y Automática. (2000). *Manual de referencia de Matlab & Simulink*. Universidad Pontificia Comillas.
- [13] *Field Wiring and Noise Considerations for Analog Signals*. <http://www.ni.com/white-paper/3344/en>
- [14] Data Acquisition Toolbox. Ayuda de Matlab.

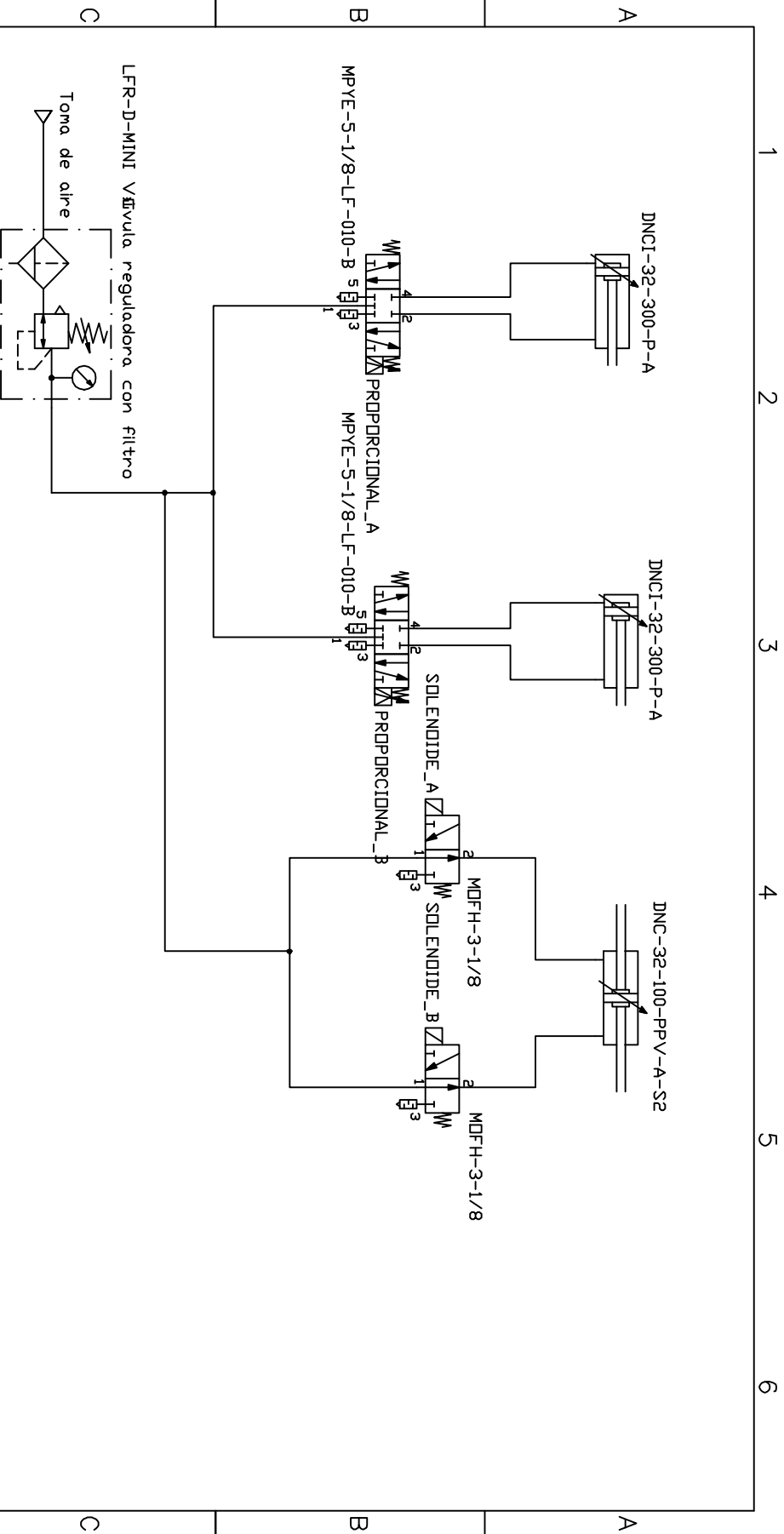
ANEXO



Marca	Denominación del componente
LFR-D-MINI	Válvula reguladora con
Toma de aire	Unidad de mantenimiento
DNCI-32-300-P-A	Fuente de aire comprimido
DNCI-32-300-P-A	Cilindro doble efecto
DNCI-32-100-PPV-A-S2	Cilindro doble efecto
DNCI-32-100-PPV-A-S2	Cilindro de doble efecto y doble vástago, con amort
MOFH-3-1/8	Válvula de 3/n vías
MOFH-3-1/8	Válvula de 3/n vías
MPYE-5-1/8-LF-010-B	Válvula de 5/n vías
MPYE-5-1/8-LF-010-B	Válvula de 5/n vías
SMAT	Transmisor de posich
Convertidor DADE-MVC	Convertidor de tensh
Convertidor DADE-MVC	Convertidor de tensh



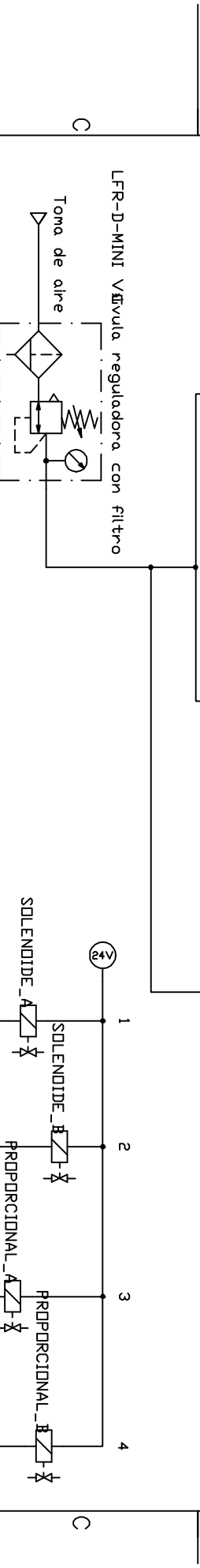
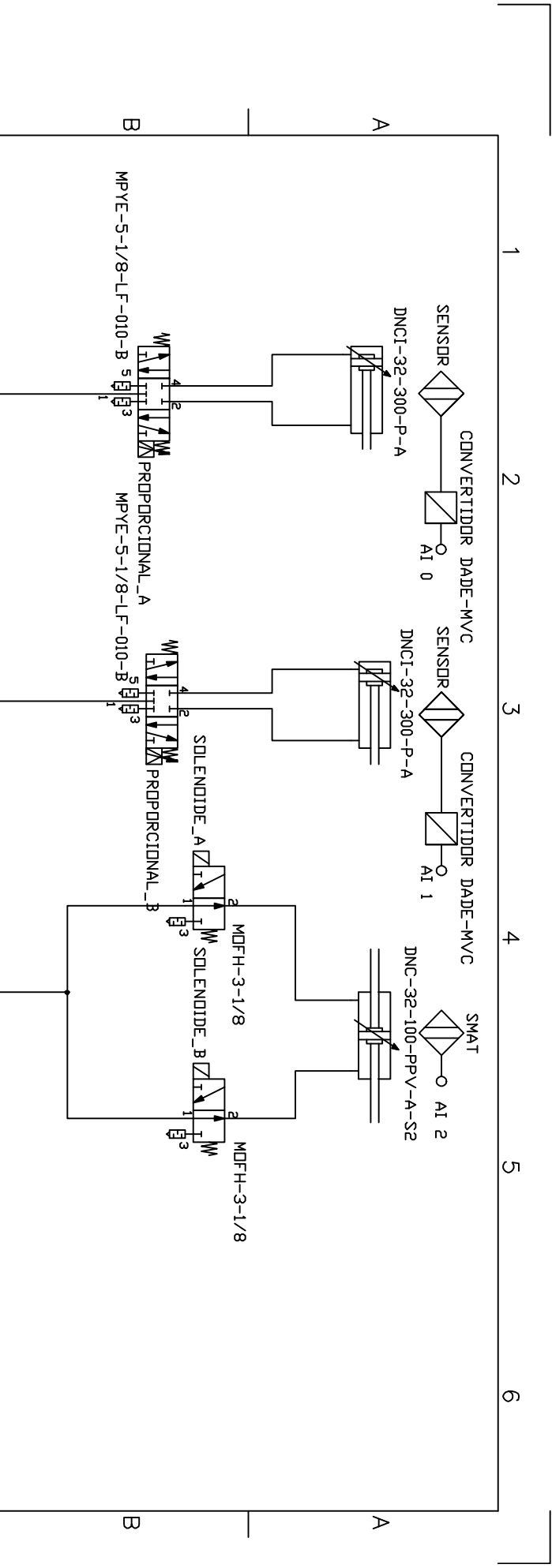
Autor: Alejandro Rosillo Meseguer	
ITI, especialidad electronica industrial	
Dpto: Ingenieria de Sistemas y Automatica	
Esquema electrico	Anexo A.1



Marca	Denominación del componente	
LFR-D-MINI	Válvula reguladora con filtro	Unidad de mantenimiento
Toma de aire		Fuente de aire comprimido
DNCI-32-300-P-A		Cilindro doble efecto
DNCI-32-300-P-A		Cilindro doble efecto
DNC-32-100-PPV-A-S2		Cilindro de doble efecto y doble vástago, con amortiguador
MDFH-3-1/8		Válvula de 3/n vías
MDFH-3-1/8		Válvula de 3/n vías
MPYE-5-1/8-LF-010-B		Válvula de 5/n vías
MPYE-5-1/8-LF-010-B		Válvula de 5/n vías
SMAT		Transmisor de posición
Convertidor JADE-HVC		Convertidor de tensión
Convertidor JADE-HVC		Convertidor de tensión

Autor: Alejandro Rosillo Meseguer	
ITI, especialidad electrónica industrial	
Dpto: Ingeniería de Sistemas y Automática	
Esquema neumático	

Anexo A.2



Marca	Denominación del componente
LFR-D-MINI	Valvula reguladora con
Toma de aire	Unidad de mantenimiento
DNCI-32-300-P-A	Fuente de aire comprimido
DNCI-32-300-P-A	Cilindro doble efecto
DNC-32-100-PPV-A-S2	Cilindro de doble efecto y doble vástago, con amort
MPYE-5-1/8-LF-010-B	Valvula de 3/n vias
MPYE-5-1/8-LF-010-B	Valvula de 3/n vias
MPYE-5-1/8-LF-010-B	Valvula de 5/n vias
SMAT	Transmisor de posich
Convertidor DADE-MVC	Convertidor de tensh
Convertidor DADE-MVC	Convertidor de tensh

Autor: Alejandro Rosillo Meseguer	
ITI, especialidad electronica industrial	
Dpto: Ingenieria de Sistemas y Automatica	
Esquema electrico y neumatico	Anexo A.3

ANEXO A.4:
PROBLEMAS Y SU RESOLUCIÓN.

A.4.1. Introducción.

El anexo A.4 recopila los distintos problemas que han ido surgiendo durante el desarrollo del presente proyecto, así como las soluciones adoptadas en cada caso.

A.4.2. Problemas y resolución.

A.4.2.1. Pérdida del interruptor electrónico.

Durante la revisión inicial de los componentes del sistema se observó que el circuito que actuaba como interruptor electrónico para referenciar los cilindros *DNCI* no estaba. Se propuso entonces, en lugar de construir otro circuito de las mismas características, utilizar un dispositivo convertidor de tensión continua que transformara los 5 V que la tarjeta de adquisición de datos era capaz de enviar a los 24 V que requerían los terminales de referencia y calibración de los convertidores *DADE*.

Se pidió a la empresa *RS* el dispositivo *DC/DC Converter RN-0524S* de los fabricantes *RECOM*, que tenía las características expuestas en el apartado 2.11 del capítulo de componentes.

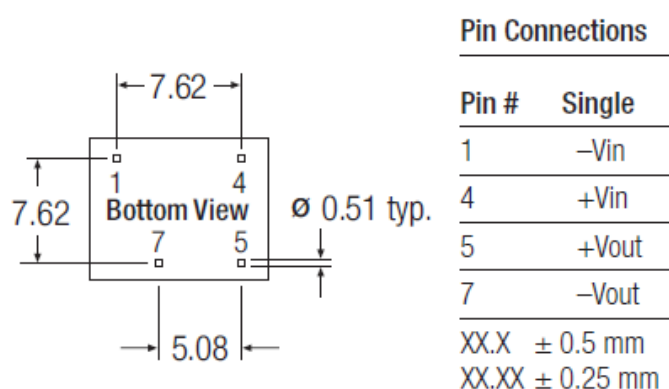


Fig. A.4.1. Terminales del dispositivo *RN-0524S*

Una vez conectado y soldado correctamente de acuerdo a las especificaciones de la hoja de características (Figura A.4.1), el circuito quedó del siguiente modo Figura (A.4.2):

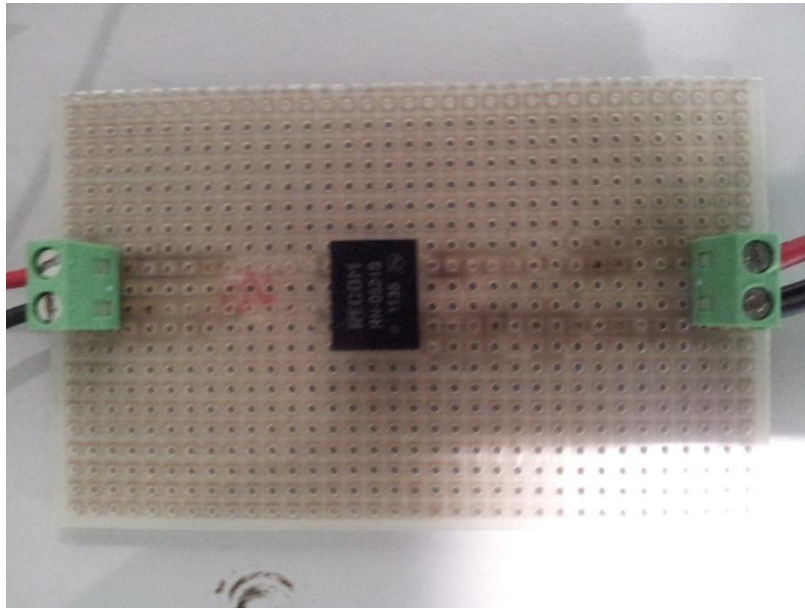


Fig. A.4.2. Circuito convertidor de tensión (5 a 24 V)

A.4.2.2. Cierre incorrecto de válvulas.

Cuando se trató de solucionar el problema de la zona muerta en los sensores de los cilindros se observó que, aunque las válvulas proporcionales no deberían permitir el paso de aire a una tensión de 5V, en realidad sí dejaban pasar una cierta cantidad que desplazaba al cilindro en uno u otro sentido.

Para solucionar esta situación se realizaron diversas pruebas con diferentes valores de tensión para averiguar cuál de ellos cerraba las válvulas por completo. Se comprobó que estos valores eran distintos para una y otra válvula. Además, la válvula del cilindro derecho cerraba a diferentes valores de tensión según éste se encontrara por encima o por debajo del 70% de su recorrido. Todo ello obligó a crear una serie de modificaciones en el código de control del robot para incluir estas pequeñas diferencias entre ambos actuadores.

A.4.2.3. Ruido de válvulas.

Al conectar la fuente de alimentación se escucha un molesto zumbido procedente de las válvulas proporcionales *MPYE*, que desaparece en cuanto éstas conmutan. Por ello se creó un código muy simple en *Matlab* cuyo único objetivo era asignar un valor de tensión a estas válvulas para eliminar la vibración.

A.4.2.4. Pérdida de presión.

Al conectar el aire comprimido, en algunas ocasiones el regulador de presión indicaba que la presión era nula, la atmosférica. Se verificó que el sistema perdía presión por el conducto de entrada a este regulador, por lo que se procedió a sellar el tubo de admisión con teflón. No obstante, el problema persistía, aunque con menor asiduidad. Realizando algunas pruebas se observó que esta presión volvía a su régimen normal en cuanto las válvulas *MOFH*, de control todo-nada, conmutaban. Cabía esperar entonces que si estas válvulas se encontraban inactivas el aire salía al exterior, por lo que se creó un código en *Matlab* encargado de activar estas válvulas sin necesidad de desplazarlas para cerrar los conductos del aire al exterior a través de ellas.

A.4.3. Problemas no solventados.

A.4.3.1. Problemas con la estructura.

Tal y como se ha comentado en el capítulo de desarrollos futuros, la estructura que alberga al robot es muy poco rígida. La placa del lateral derecho tuvo que ser sustituida en proyectos anteriores para que no interfiriese con el movimiento de los cilindros, lo que originó que la estructura se encuentre libre por dicho flanco. Así, al realizar las pruebas de dibujo sobre un papel, la placa posterior sobre la que el lápiz ejercía presión no estaba bien alineada respecto del efector del robot. Del mismo modo, la estructura del robot también presenta una cierta holgura que acentúa aún más esta situación.

Se observaba que las figuras que mostraba *Matlab* a través de las gráficas no siempre correspondían al dibujo realizado sobre el papel debido a esta diferencia. Incluso en ocasiones la distancia entre el papel y el lápiz se hacía patente, de forma que no se escribía nada. Además, control todo-nada del efector final impedía solventar este problema por medios automáticos.

Como medida provisional se fijó manualmente la placa posterior de la estructura que rodea al robot y se pudieron realizar las representaciones gráficas sobre el papel.

A.4.3.2. Problemas con la tensión que llega a la válvula izquierda.

Este problema surgió a mitad del desarrollo del proyecto y se caracterizó por su intermitencia, lo que permitió finalmente realizar las pruebas que se pretendían.

La primera observación fue que el cilindro izquierdo se atrancaba. Su movimiento en algunas ocasiones no era fluido, sino que le costaba mucho tiempo arrancar y cuando lo hacía era de forma brusca. Para tratar de averiguar dónde estaba el fallo se procedió a limpiar el vástago del cilindro y a limpiar la válvula con aire comprimido a baja presión. Puesto que el problema persistía, en la bornera de conexión a la tarjeta de adquisición de datos se intercambiaron los terminales correspondientes a la válvula izquierda y derecha para verificar si el fallo era propio de la válvula o del cilindro. Al realizar este proceso era el cilindro derecho el que funcionaba del mismo modo que antes el opuesto, por lo que se concluyó que el problema estaba en la válvula.

Uno de los aspectos más importantes de estas válvulas es que requieren para su conmutación una diferencia de presión de 1.5 bar y un valor de tensión superior a 5.6 V para desplazar el cilindro. Se llevó a cabo un control en lazo abierto del cilindro izquierdo, introduciendo a la válvula distintos valores de tensión para comprobar si ésta tensión era recibida o no. Efectivamente, el voltaje que llegaba a la válvula en las situaciones de fallo nunca era superior a 5.4 V. Con este nivel de tensión el cilindro no podía desplazarse.

El problema entonces parecía estar en la tarjeta de adquisición de datos, que no sacaba por el pin correspondiente la tensión indicada. Sin embargo, para asegurarnos de este hecho se desconectó la válvula y se midió los niveles de tensión que proporcionaba la tarjeta de adquisición de datos en vacío. De este modo el voltaje que llegaba a la bornera siempre era el indicado, sin que hubiera ningún tipo de perturbación.

Tras esta serie de experimentos se llegó a la conclusión de que es el cable de la válvula el que, de algún modo, ofrece una carga resistente que impide que a ésta le llegue el valor de tensión enviado por la tarjeta. Debido a falta de tiempo no se pudo sustituir el cable por otro, sino que se procedió a su evaluación y acondicionamiento. Se cortó y se dejó un mayor apantallamiento del mismo. Además, se suprimieron los empalmes de modo que estuviera conectado directamente a la bornera.

El problema no se solventó por completo, pero sí que tenía lugar con mucha menos frecuencia, lo que permitió finalizar el proyecto sin demasiadas interrupciones por esta causa.

ANEXO A.5:
CÓDIGO DE *MATLAB*.

A.5.1. referencia.m

```
%Crea objeto de i/o digital y línea.
dio = digitalio('nidaq',1);
addline(dio,3,0,'out');

%Saca 5V al cable gris para establecer punto referencia
putvalue(dio,1)
tic
for i=1:50000^2    %Contador de duración 6 seg. aprox.
end
%Saca 0V al cable gris para finalizar referencia
putvalue(dio,0)
toc

delete(dio)
clear dio
```

A.5.2. parametros.m

```
a=60; %distancia entre apoyos (cm)
b=20; %longitud del eslabón de unión (cm)
factor=0.5; %factor numérico que sitúa el efector sobre el eslabón
sit=b*(factor); %situación del efector sobre el eslabón de unión (cm)
tita3=2*pi/3; %ángulo entre el cilindro 1 y el eslabón unión (radianes)
l=50; %longitud del cilindro (cm)
```

A.5.3. cdir.m

```
%programacion del espacio de trabajo del robot

function [xp,yp]=cdir(q1,q2);

format long

parametros

upa=sqrt((l+q1)^2+(b^2)-(2*(l+q1)*b*cos(tita3)));
upb=(l+q2);
tita2=pi-(tita3);
xa=0;
ya=0;
xb=a;
yb=0;
%2*(xb-xa)*x+2*(yb-ya)*y+R-S=0;
R=(xa^2)+(ya^2)-upa^2;
S=(xb^2)+(yb^2)-upb^2;
M=(ya-yb)/(xb-xa);
N=(1/2)*((S-R)/(xb-xa));
%p*(y^2) + q*y + r = 0
p=(M^2)+1;
q=2*((M*N)-(M*xa)-ya);
r=(N^2)-(2*N*xa)+R;
z=(q^2-4*p*r);
if z>=0
    yd=(-q+sqrt(q^2-4*p*r))/(2*p);
    y2=(-q-sqrt(q^2-4*p*r))/(2*p);
    xd=M*yd+N;
    x2=M*y2+N;
```

```

    %calcula del angulo tita2
    d=sqrt((xd^2)+(yd^2));
    alfa=atan(yd/xd);
    beta=acos((b^2-(l+q1)^2-d^2)/(-2*d*(l+q1)));
    tita2=alfa+beta;
    xc=(l+q1)*cos(tita2);
    yc=(l+q1)*sin(tita2);
    if yc>=0
        %calcula de la posicion del actuador
        fi=tita2+tita3-pi;
        xp=xc+(sit)*cos(fi);
        yp=yc+(sit)*sin(fi);

        disp('disposicion fisica imposible')
    end
end
end

```

A.5.4. *cinva.m*

```

function [q1,q2]=cinva(px,py);
format long
parametros

q1=0;
q2=0;

%parametros geometricos necesarios
alfa=atan(py/px);
d=sqrt((py^2)+(px^2));
beta=asin((sit*sin(tita3))/d);
tita2=alfa+beta;
fi=tita2+tita3-pi;

%calcula del punto C para calcular d1
xc=px-sit*cos(fi);
yc=py-sit*sin(fi);
m=sqrt((yc^2)+(xc^2));
d1=m-l;

%calcula del punto D para calcular d2
xd=px+sit*cos(fi);
yd=py+sit*sin(fi);
n=sqrt((yd^2)+(a-xd)^2);
d2=n-l;
q1=d1;
q2=d2;

```

A.5.5. PID.m

```
function [ai, ao, dio] = PID()
clear all

%NOMENCLATURA:
% 1 : cilindro izquierdo
% 2 : cilindro derecho
% 3 : actuador horizontal

%declara las variables
global S_REFERENCIA1;
global S_REFERENCIA2;
global S_REFERENCIA3;
global SALIDA1;
global SALIDA2;
global INTEGRAL;
global N_MUESTRA;
global S_CONTROL1;
global S_CONTROL2;
global S_CONTROL3;

%Asignación dinámica de las variables
S_CONTROL1=[];
S_CONTROL2=[];
S_CONTROL3=[];
S_REFERENCIA1=[];
S_REFERENCIA2=[];
S_REFERENCIA3=[];
SALIDA1=[];
SALIDA2=[];

%inicia N_MUESTRA e INTEGRAL
N_MUESTRA = 1;
INTEGRAL = [0 0];

%Driver National Instrument: nidaq
%crea los objetos ai y ao y asigna los canales
ao = analogoutput('nidaq',1);
ai = analoginput('nidaq',1);
dio = digitalio('nidaq',1);
chans_o = addchannel(ao,0:1);
chans_i = addchannel(ai,0:1);
chans_d = addline(dio,0:1,'out');

%Ajusta los rangos de las entradas a los de los sensores
set(chans_i,'SensorRange',[0 10]);
set(chans_i,'InputRange',[0 10]);
set(chans_i,'UnitsRange',[0 10]);

%ajusta los rangos de salida a los de los actuadores
set(chans_o,'OutputRange',[0 10]);
set(chans_o,'UnitsRange',[0 10]);

% Frecuencia de muestreo (Hertzios)
set(ai,'SampleRate',30);
ai.SamplesAcquiredFcnCount = 1;
ai.InputType='Differential';
% Función que implementa el algoritmo de control
ai.SamplesAcquiredFcn = {@callback_PID, ao, dio};
%Numero de muestras que se van a adquirir en cada disparo "Trigger"
set(ai,'SamplesPerTrigger',inf);
set(ai,'triggerrepeat',inf);

start(ai);
tic
```

A.5.6. *callback_PID.m*

```
function callback_PID(obj,event, ao, dio)

%Declaración de variables
global REFERENCIA1;
global REFERENCIA2;
global REFERENCIA3;
global N_MUESTRA;
global INTEGRAL;
global S_CONTROL1;
global S_CONTROL2;
%global S_CONTROL3;
global S_REFERENCIA1;
global S_REFERENCIA2;
%global S_REFERENCIA3;
global SALIDA1;
global SALIDA2;
global U;
global xp;
global yp;
global d1;
global d2;
global sal;
persistent E_k1;
global X;
global Y;
global EJEX;
global EJEY;
global REFERENCIA11;
global REFERENCIA12;

%E_k1 es el error en la muestra anterior para N_MUESTRA==1 valdra 0
if (N_MUESTRA == 1)
    E_k1 =[0 0];
end

%Situa los valores de los distintos canales de ai en la variable sal
sal=getsample(obj);

%Establecimiento de las coordenadas de la figura

r=4; %radio
xc=35; %coordenada 'x' del centro
yc=55; %coordenada 'y' del centro
n=4; %número de intervalos o lados
N=10; %número de puntos para trazar cada lado

[X,Y]= poligono(xc,yc,r,n,N);

%Guardamos las coordenadas de referencia en dos vectores

inimues=1;
finmues=100;
INC=1;
per=40; %periodo de muestreo de cada muestra
for i=1:length(X)

    EJEX(1,inimues:finmues)=X(1,INC);
    EJEY(1,inimues:finmues)=Y(1,INC);

    if inimues==1
        inimues=inimues+finmues;
    else inimues=inimues+per;
    end
    finmues=finmues+per;
    INC=INC+1;
```

```

end

REFERENCIA1=0;
REFERENCIA2=0;
REFERENCIA3=0;

%Cálculo de la posición de los cilindros a partir de las coordenadas del
%actuador

[REFERENCIA11(N_MUESTRA),REFERENCIA12(N_MUESTRA)]=cinv(EJEX(N_MUESTRA),EJEY(N_
MUESTRA));

REFERENCIA1(N_MUESTRA)=(REFERENCIA11(N_MUESTRA)*10)/30;
REFERENCIA2(N_MUESTRA)=(REFERENCIA12(N_MUESTRA)*10)/30;

REFERENCIA3(N_MUESTRA)=1;

%valores de los PID, podemos ajustarlo para obtener una respuesta diferente

Kp(1)=2;
Ki(1)=0.001;
Kd(1)=0.02;

Kp(2)=2.5;
Ki(2)=0.001;
Kd(2)=0.025;

%IMPLEMENTACION DEL CILINDRO 1

SALIDA1(N_MUESTRA) = sal(1);
S_REFERENCIA1(N_MUESTRA)=REFERENCIA1(N_MUESTRA);

%Posicion
Y_k(1) = sal(1);
>Error
E_k(1) = REFERENCIA1(N_MUESTRA)-Y_k(1);
Integral
INTEGRAL(1) = INTEGRAL(1) + E_k(1);

%Señal de control
U(1) = Kp(1) * E_k(1) + Ki(1) * INTEGRAL(1) + Kd(1) * (E_k(1) -
E_k1(1))/(1/30);

%para ver si eliminamos la zona muerta
if REFERENCIA1(N_MUESTRA)-0.12<Y_k(1)&&REFERENCIA1(N_MUESTRA)+0.12>Y_k(1)
U(1)=-0.1;
else
U(1) = Kp(1) * E_k(1) + Ki(1) * INTEGRAL(1) + Kd(1) * (E_k(1) -
E_k1(1))/(1/30);
end

%La señal de control oscila en torno al cero
%Saturacion que mantiene la señal de control en valores entre -5 y 5
if U(1)>5
U(1)=5;
elseif U(1)<-5
U(1)=-5;
end

%Sumamos 5 para que el valor que introducimos al actuador sea entre 0
%y 10
U(1) = U(1)+5;

%Si la señal esta saturada evitamos que INTEGRAL aumente su valor
if U(1)>=6 || U(1)<=4
INTEGRAL(1) = INTEGRAL(1) - E_k(1);
end

```



```

%Mantenemos la señal de control en un rango estrecho para evitar
%movimientos bruscos
if U(1)>5.7
U(1)=5.7;
elseif U(1)<4.1
U(1)=4.1;
end

%IMPLEMENTACION DEL CILINDRO 2

SALIDA2(N_MUESTRA) = sal(2);
S_REFERENCIA2(N_MUESTRA)=REFERENCIA2(N_MUESTRA);

%Posicion
Y_k(2) = sal(2);
>Error
E_k(2) = REFERENCIA2(N_MUESTRA)-Y_k(2);
%Integral
INTEGRAL(2) = INTEGRAL(2) + E_k(2);

%Señal de control

%para ver si eliminamos la zona muerta
if REFERENCIA2(N_MUESTRA)-
0.12<Y_k(2) &&REFERENCIA2(N_MUESTRA)+0.12>Y_k(2) &&REFERENCIA2(N_MUESTRA)<=6
U(2)=0.08;
else if REFERENCIA2(N_MUESTRA)-
0.12<Y_k(2) &&REFERENCIA2(N_MUESTRA)+0.12>Y_k(2) &&REFERENCIA2(N_MUESTRA)>6
U(2)=0.1;
else U(2) = Kp(2) * E_k(2) + Ki(2) * INTEGRAL(2) + Kd(2) * (E_k(2) -
E_k1(2))/(1/30);
end
end

%La señal de control oscila en torno al cero
%Saturacion que mantiene la señal de control en valores entre -5 y 5
if U(2)>5
U(2)=5;
elseif U(2)<-5
U(2)=-5;
end

%Sumamos 5 para que el valor que introducimos al actuador sea entre 0
%y 10
U(2) = U(2)+5;

%Si la señal esta saturada evitamos que INTEGRAL aumente su valor
if U(2)>=6 || U(2)<=4
INTEGRAL(2) = INTEGRAL(2) - E_k(2);
end

%Mantenemos la señal de control en un rango estrecho para evitar
%movimientos bruscos
if U(2)>6
U(2)=6;
elseif U(2)<4.2
U(2)=4.2;
end

% IMPLEMENTACIÓN DEL CILINDRO 3

if REFERENCIA3(N_MUESTRA)==1
A = 0;
B = 1;
else
end

```

```

%ha finalizado la implementación de los cilindros

S_CONTROL1(1,N_MUESTRA)= U(1);
S_CONTROL2(1,N_MUESTRA)= U(2);

putsample(ao, [U(1) U(2)]);
putvalue(dio,[A B]);

%Calculamos la posición real del actuador para representación gráfica

d1(N_MUESTRA)=(SALIDA1(N_MUESTRA)*30)/10;
d2(N_MUESTRA)=(SALIDA2(N_MUESTRA)*30)/10;

[xp(N_MUESTRA),yp(N_MUESTRA)]=cdir(d1(N_MUESTRA),d2(N_MUESTRA));

% Finalización del programa
if N_MUESTRA == numel(EJEX)
    finaliza(ai,ao,dio);
end

%Actualización del error en el intervalo anterior
E_k1 = E_k;

%Pasamos a la muestra siguiente
N_MUESTRA = N_MUESTRA + 1;

```

A.5.7. *finaliza.m*

```

function finaliza(ai, ao, dio)
toc

%Declaracion de variables
global SALIDA1;
global S_REFERENCIAL1;
global S_CONTROL1;
global S_REFERENCIA2;
global SALIDA2;
global S_CONTROL2;
global xp;
global yp;
global X;
global Y;

%Devuelve el cilindro a la posición inicial
putsample(ao,[4,4]);
putvalue(dio,[1 0]);

%Para y elimina los objetos ai, ao y dio
stop(ai);
stop(ao);
delete(ai);
delete(ao);
stop(dio);
delete(dio);

%Muestra por pantalla los resultados del cilindro 1
figure()
plot(S_REFERENCIAL1,'r');
hold on;

plot(S_CONTROL1,'g');
hold on;

```

```

plot(SALIDA1);
hold on;
title('CILINDRO 1')
legend('Referencia','Señal de control','Posición')
xlabel('Número de muestras')
ylabel('Voltaje (V)')

%muestra los resultados del cilindro 2
figure()
plot(S_REFERENCIA2,'r');
hold on;
plot(S_CONTROL2,'g');
hold on;
plot(SALIDA2,'b');
hold on;
title('CILINDRO 2')
legend('Referencia','Señal de control','Posición');
xlabel('Número de muestras')
ylabel('Voltaje (V)')

%representamos el punto de actuacion del actuador
figure();
axis([-20,80,0,80])
xlabel('distancia horizontal (cm)');
ylabel('distancia vertical (cm)');
plot(xp,yp,'r')
axis equal
hold on

%Representación de la figura geométrica teórica
plot(X,Y,'b');

```

A.5.8. *poligono.m*

```

function [x,y]= poligono(xc,yc,r,n,N)

%Proporciona las coordenadas (x,y) para construir un polígono regular

%xc: coordenada 'x' del centro
%yc: coordenada 'y' del centro
%r: radio del polígono regular
%n: número de lados del polígono
%N: número de puntos para trazar cada lado

%inicializamos los vectores x e y con valor nulo
x=zeros(1,(N-1)*n+1);
y=zeros(1,(N-1)*n+1);

%almacena en dos vectores los vértices de la figura
for j=1:N-1:(N-1)*n+1
    ang(j)=2*pi*(j-1)/((N-1)*n);
    x(j)=xc+r*cos(ang(j));
    y(j)=yc+r*sin(ang(j));
end

%divide cada lado en N puntos (N-1 intervalos)
for j=2:1:(N-1)*n
    if x(j)==0 && y(j)==0
        x(j-1:j+N-2)=linspace(x(j-1),x(j+N-2),N);
        y(j-1:j+N-2)=linspace(y(j-1),y(j+N-2),N);
    else
        end
end
end

```